



Universidad del Azuay

Facultad de Ciencias de la Administración

Escuela de Ingeniería de Sistemas y Telemática

**CLASIFICACIÓN DE SÍLABOS ACADÉMICOS
EN BASE A REDES NEURONALES DE
APRENDIZAJE PROFUNDO**

Trabajo de graduación previo a la obtención del título de
Ingeniero en Sistemas y Telemática

Autor:

Juan Marcelo Alvarado Carrera

Director:

Ing. Marcos Orellana.

Cuenca – Ecuador

2018

DEDICATORIA

A mi esposa

Por haberme incitado siempre a culminar mis metas
y apoyarme en las etapas de la vida que hemos
vivido juntos. Mostrando una perseverancia en
cualquier situación que se nos presente con la mejor
actitud y fuerza.

AGRADECIMIENTO

A Dios por guiarme en mi vida
Agradezco al apoyo de mis padres y mi hermano por
contribuir a mi desarrollo personal y educativo, así
como a mi director y codirector de tesis quienes me
han apoyado en la realización de este proyecto que
constituye una parte esencial de mi parte académica.
Sin olvidar a mi tío abuelo que siempre me apoyo en
mi vida personal.

RESUMEN:

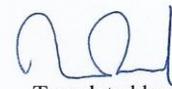
A fin de facilitar la homologación de créditos entre las universidades del país, se requiere automatizar la clasificación de sílabos en áreas y subáreas. Esta investigación tiene como objetivo clasificar los sílabos mediante la técnica de redes neuronales conectadas de aprendizaje profundo, a través de una combinación de número de capas, funciones de activación, tamaños y épocas de entrenamiento. Este modelo fue comparado con respecto a algoritmos basados en *Support Vector Machine (SVM)*, *Naive Bayes* y Árboles de decisión. Los resultados demostraron que el modelo de aprendizaje profundo propuesto fue superior en 1.4% con respecto a *Naive Bayes*, 6.2% con respecto a *SVM* y 7.2% con respecto a Árboles de decisión.

Palabras clave: aprendizaje profundo, *SVM*, árbol de decisión, parámetros, Naive Bayes.

ABSTRACT

It was required to automate the classification of syllables in areas and sub-areas in order to facilitate the homologation of credits among the universities of the country. The objective of this research was to classify the syllables using the technique of connected neural networks of deep learning through a combination of number of layers, activation functions, sizes and times of training. This model was compared with algorithms based on Support Vector Machine (SVM), Naive Bayes and Decision Trees. The results showed that the proposed deep learning model was superior in 1.4% with respect to Naive Bayes, 6.2% with respect to SVM and 7.2% with respect to Decision trees.




Translated by
Ing. Paul Arpi

ÍNDICE

Índice de contenido

1. INTRODUCCIÓN	1
1.2 <i>Conceptos generales del aprendizaje automático</i>	3
1.2.1 Generalización	3
1.2.2 Sobreajuste	3
1.2.3 Ajuste insuficiente	3
1.2.4 Validación cruzada	3
1.3 <i>Aprendizaje automático en la clasificación de texto</i>	4
1.4 <i>Aprendizaje profundo</i>	5
1.4.1 Tipos de redes de Aprendizaje Profundo	5
1.4.1.1 Redes Convolucionales	5
1.4.1.2 Redes Recurrentes	5
1.4.1.3 Redes Totalmente Conectadas	6
1.4.2 Parámetros del aprendizaje profundo	6
1.4.2.1 Optimización de la gradiente	6
1.4.2.2 Funciones de activación neuronal.	7
1.4.2.3 Regularización	8
1.4.2.4 Normalización por lotes (<i>BatchNormalization</i>)	9
1.4.2.5 Épocas (<i>epochs</i>)	9
1.4.2.6 Lotes (<i>batch_size</i>)	9
1.5 <i>Evaluación de algoritmos</i>	9
2. MATERIALES Y MÉTODOS	11
2.1 <i>Materiales</i>	11
2.1.1 Datos	11
2.1.2 Software	12
2.1.3 Entorno de pruebas	12
2.2 <i>Método</i>	13
2.2.1 Datos	14
2.2.2 Construcción del modelo computacional de aprendizaje profundo.	16
2.2.2.1 Estructura del modelo	16
2.2.2.2 Optimización de parámetros	17
2.2.2.3 Evaluación de algoritmos	19
2.2.2.4 Repetición de datos.	19
2.2.2.5 La exactitud como medida de evaluación de algoritmos	19

3. RESULTADOS	20
3.1 <i>Construcción del modelo computacional de aprendizaje profundo</i>	20
3.1.1 Estructura del modelo	20
3.1.2 Optimización de parámetros	20
Tabla 6	21
3.1.3 Evaluación de algoritmos	22
3.1.4 Repetición de datos	23
3.1.5 La exactitud como medida de evaluación de algoritmos.	23
4. DISCUSIÓN Y CONCLUSIONES	25
BIBLIOGRAFÍA	27

Índice de tablas

Tabla 1	Función de activación	8
Tabla 2	Conjunto de sílabos preclasificados para el entrenamiento de los algoritmos	11
Tabla 3	Parámetros para la experimentación	17
Tabla 4	Ejecución de la primera cuadrilla.	20
Tabla 5	Ejecución de la segunda cuadrícula	21
Tabla 6	Ejecución de la tercera cuadrícula	21
Tabla 7	Aumento de número de capas.	22
Tabla 8	Tabla comparativa de algoritmos en la clasificación de sílabos.	22
Tabla 9	Comparación de algoritmos con clases balanceadas y no balanceadas.	24

Índice de figuras

Ilustración 1	Procesamiento de datos	13
Ilustración 2	Selección de elementos para estructurar el modelo.	14
Ilustración 3	Gráfico de aumento de datos.	23

1. INTRODUCCIÓN

La clasificación se define como el acto de dividir un conjunto de elementos en clases a partir de un criterio (RAE, 2017), esta acción permite operar con dichos elementos para propósitos definidos en cada ámbito de su implementación. Si se lo realiza de forma manual consiste en un proceso tedioso y sujeto al criterio del evaluador en turno, pues como menciona Bowker & Star (2000) al clasificar “se valoriza algún punto de vista y se silencia otro” dependiendo de la interpretación que se defina para designar una categoría, es por eso que automatizar este proceso es esencial. Una forma de hacerlo es utilizando técnicas de aprendizaje automático (Kotsiantis, Zaharakis, & Pintelas, 2007).

Uno de los elementos en los que se ha aplicado la clasificación consiste en los sílabos académicos. Un caso en particular surge debido a la circulación de estudiantes entre las diferentes universidades del país, pues con esta situación nace la necesidad de clasificar sílabos en áreas y subáreas de conocimiento determinadas por la UNESCO con el fin de facilitar la comparación de la similitud de contenido académico para el proceso de homologación de créditos. La revisión de literatura muestra lo descrito a continuación acerca de la temática de la investigación.

En el área de clasificación de sílabos se ha trabajado en aspectos como: a) determinar cuáles son los efectos que tienen los componentes de los programas con respecto al género y las instituciones utilizando búsquedas avanzadas de Google (Doolittle & Lusk, 2007), b) proveer una idea clara de los sílabos definiendo su contenido mediante una clasificación con un enfoque orientado al tema mediante el algoritmo *Support Vector Machine* (SVM) (Ota & Mima, 2011), c) implementar una clasificación basada en la definición de un modelo semántico (Chung & Kim, 2016), d) y en la filtración de ruido en motores de búsqueda clasificando entradas en sílabos válidos con su nivel de completitud utilizando *Naive Bayes* (NB) y *SVM* (Yu et al., 2007).

En el área de clasificación de texto en general se han usado a) enfoques bayesianos, siguiendo el Teorema de proyección PDF de Baggenstoss. Este teorema se utiliza para construir archivos PDF en el espacio de datos sin formato, fue aplicado en un conjunto de datos no balanceado (Tang, He, Baggenstoss, & Kay, 2016). De igual forma, el uso de *SVM* se ha aplicado en la aceleración de búsquedas (Abdiansah & Wardoyo, 2015). En el

sector comercial existe un software llamado *fastTex* de la compañía Facebook que tiene un gran rendimiento aplicando *SVM* (Joulin, Grave, Bojanowski, & Mikolov, 2016).

Sin embargo, existe una técnica de aprendizaje automático que ha demostrado su potencial en la determinación de emociones y tareas relacionados con textos e imágenes (Deng, Yu, & others, 2014), se trata del aprendizaje profundo. Éste se implementa mediante arquitecturas recurrentes, convolucionales, totalmente conectadas e híbridas. Las estructuras convolucionales se diseñaron para procesar datos representados como matrices múltiples, las recurrentes se conforman de un modelo secuencial neuronal que mantiene un historial de los elementos transitados (LeCun, Bengio, & Hinton, 2015), las totalmente conectadas, son redes que contienen más de una capa oculta entre sus entradas - salidas (Hinton et al., 2012) y las híbridas, constituyen una combinación de las estructuras antes mencionadas.

Dentro de la literatura se encontró a) el uso de una red neuronal convolucional recurrente para clasificar texto (Lai, Xu, Liu, & Zhao, 2015) b) una red convolucional para recategorizar pares de texto corto buscando su similitud con un entrenamiento supervisado (Severyn & Moschitti, 2015) y c) una red recurrente para categorizar texto, comparando resultados de clasificación con algoritmos de *SVM*, redes totalmente conectadas y convolucionales (Johnson & Zhang, 2014).

Bajo este panorama, se puede observar que la clasificación de texto usando sílabos no se ha realizado bajo el criterio de las áreas y subáreas del conocimiento designadas por la UNESCO y que no existe una aplicación directa del aprendizaje profundo en esta tarea educativa, ni antecedentes que evalúen técnicas de aprendizaje automático en contraste con el aprendizaje profundo en este contexto. Es por ello, que se va a clasificar los planes de estudio académicos en base a redes neuronales de aprendizaje profundo.

Corresponde como primer paso hacer una revisión de los conceptos que organizan esta línea de investigación partiendo por la inteligencia artificial, el aprendizaje automático y el aprendizaje profundo. Lo que permitirá seleccionar los algoritmos de aprendizaje automático para la experimentación, así como conocer el funcionamiento, arquitectura y parámetros a optimizar en el aprendizaje profundo.

La inteligencia artificial es un campo que examina como proveer a máquinas la habilidad de aprender y resolver problemas. Dentro de este campo, se desarrolló el

aprendizaje automático que estudia la generación de algoritmos para predecir (Ongsulee, 2017). De esta generación de algoritmos nace el Aprendizaje Profundo.

El aprendizaje profundo se puede definir como el estudio de redes neuronales artificiales que poseen más de una capa oculta (Ongsulee, 2017). Estas capas ocultas llamadas abstracciones tiene un conjunto de parámetros que se pueden optimizar (LeCun et al., 2015). Como contexto general, a continuación, se presenta una serie de conceptos generales del aprendizaje automático.

1.2 Conceptos generales del aprendizaje automático

1.2.1 Generalización

La generalización dentro del aprendizaje automático es la capacidad de producir predicciones correctas en nuevas entradas de datos, los cuales no han sido parte del entrenamiento del algoritmo (LeCun et al., 2015).

1.2.2 Sobreajuste

El sobreajuste llamado *overfitting* hace referencia a la situación que surge cuando un modelo computacional se ha acoplado de sobremanera a los datos de entrenamiento y no tiene la capacidad de generalizar de manera adecuada (der Aalst et al., 2010).

1.2.3 Ajuste insuficiente

El ajuste insuficiente o llamado *underfitting* hace referencia a la situación que surge cuando un modelo tiene un comportamiento extraño al generalizar, esto se debe a un entrenamiento con datos insuficientes (der Aalst et al., 2010).

1.2.4 Validación cruzada

La validación cruzada es un proceso que consiste en entrenar los algoritmos de aprendizaje automático con un conjunto de datos y luego comparar el proceso de generalización mediante datos totalmente desconocidos, ajenos al entrenamiento para obtener una buena estimación del rendimiento del algoritmo. Debido a que en la práctica muchas veces se tiene una cantidad limitada de datos, es conveniente dividir estos en porcentaje de entrenamiento y validación, esta división debe ser aleatoria (Arlot, Celisse, & others, 2010).

1.3 Aprendizaje automático en la clasificación de texto

Una vez que se ha conceptualizado algunos aspectos importantes del aprendizaje automático es necesario conocer cómo se compone. El aprendizaje automático o también llamado aprendizaje de máquina, posee un conjunto de algoritmos que se clasifican en categorías. Estas están basadas en:

Lógica: son los llamados árboles de decisión, los cuales constan de una estructura jerárquica en la que un nodo representa un atributo y cada rama un valor que el nodo puede asumir, llegando a la clasificación de un elemento en su nodo final. Algoritmos: *ID3, C4.5, CART*.

Reglas de decisión: su objetivo es que mediante un conjunto pequeño de reglas coherentes se entrenen los datos correspondientes y estos permitan descubrir las suposiciones de los datos de prueba. Algoritmos: *SCAR, DL-ESC y Ripper*.

Perceptrones: conforman un número de neuronas de entrada, capas intermedias y unidades de salida. Sus aspectos fundamentales son: funciones de entrada, activación de la unidad, arquitectura de red y el peso de cada conexión de entrada. Algoritmo: perceptrones multicapa.

Instancias: produce la clasificación comparando la similitud entre las características del vector de entrenamiento y las de pruebas, para asignar la más cercana. Son algoritmos de aprendizaje lento. Algoritmo: *KNN* o *K* vecino más cercano.

Aprendizaje estadístico: tienen un modelo de probabilidad de que una instancia pertenezca a una clase. Algoritmo: *Naive Bayes*.

Support Vector Machine (SVM): Se basa en la teoría de aprendizaje estadístico, es un algoritmo robusto que gira en torno a la noción de un margen, coloca las clases en hiperplanos, al maximizar el margen de separación entre clases obtiene menos error de generalización. Algoritmos: *LinearSVM, SVM*.

Este conjunto de categorías y su definición están tomadas de Wanjun & Xiaoguang (2010) y Kotsiantis et al (2007). De éstas las seleccionadas para la experimentación son las basadas en lógica, en aprendizaje estadístico y en *SVM*. Dentro de las basadas en aprendizaje estadístico y *SVM* la literatura destaca los algoritmos *Naive Bayes* (Yu et al., 2007; Tang et al., 2016) y *SVM* (Joulin et al., 2016; Abdiansah & Wardoyo, 2015; Doolittle & Lusk, 2007) respectivamente, y en la categoría basada en lógica, los árboles

de decisión se posicionan como el algoritmo número uno en el TOP 10 de algoritmos de clasificación según Wu et al. (2008).

1.4 Aprendizaje profundo

Una vez seleccionados los algoritmos a evaluar, se pasa a conceptualizar el aprendizaje profundo, su estructura y a determinar mediante la literatura el conjunto de parámetros a utilizar.

El aprendizaje profundo se compone de varias capas de procesamiento que aprenden de los datos, estas capas forman niveles de abstracción. Su proceso consiste en tomar los datos de entrada e introducirlos en las capas, una vez ahí estos son distorsionados de forma no lineal hasta la salida de la red donde el resultado debe ser linealmente separable. Esto se debe a que su salida debe pertenecer a una sola categoría de la lista de clasificadores (LeCun et al., 2015).

1.4.1 Tipos de redes de Aprendizaje Profundo

1.4.1.1 Redes Convolucionales

Las redes neuronales convolucionales son estructuras que se han diseñado para procesar datos representados en forma de matrices múltiples, se componen de capas convolucionales y de agrupamiento. Las capas convolucionales filtran mediante un banco de criterios en un mapa de características detectando conjunciones locales, una vez detectadas, las capas de agrupamiento fusionan características semánticamente similares (LeCun et al., 2015). Es por ello, que se han usado para determinar frases discriminatorias en el texto, captando la semántica de las oraciones (Lai, Xu, Liu, & Zhao, 2015).

1.4.1.2 Redes Recurrentes

Mikolov (2012) citado en Zaremba, Sutskever, & Vinyals (2014) las define como un modelo de secuencia neural que se destaca en el modelado de lenguaje. Estas redes mantienen en sus unidades ocultas un vector de estado con el historial de los elementos que han transitado en la secuencia, un ejemplo de su uso es la predicción de caracteres continuos de una cadena de texto (LeCun et al., 2015). Las redes recurrentes son modelos sesgados, debido a que su orden hace dominantes las palabras posteriores en contraste con las anteriores (Lai et al., 2015).

1.4.1.3 Redes Totalmente Conectadas

Una red totalmente conectada se define como una red neuronal artificial de retroalimentación o propagación hacia atrás que contiene más de una capa de unidades ocultas entre sus entradas y salidas (Hinton et al., 2012). Todas estas capas, también llamadas capas densas, están seguidas de una función de activación no lineal (K, 2018), hasta su salida que debe ser linealmente separable.

1.4.2 Parámetros del aprendizaje profundo

Luego de definir las estructuras, se procede a conceptualizar los parámetros a optimizar en la experimentación.

1.4.2.1 Optimización de la gradiente

La optimización de gradiente de una red neuronal se produce mediante el entrenamiento de descenso de gradiente estocástico. En este proceso, la máquina modifica los pesos internos de las conexiones entre capas para reducir el error, para ello, calcula un vector de gradiente, y se ajusta en su dirección opuesta buscando un mínimo donde el error promedio sea bajo (LeCun et al., 2015).

Para el cálculo de cada peso subsiguiente se utiliza la ecuación 1, en la cual W_m es el peso actual, W_{m+1} es el nuevo peso, α constituye la razón de aprendizaje, que es un valor muy pequeño que permite la convergencia hacia el mínimo, y $F'(W_m)$ es la derivada con respecto al peso anterior. Esta derivada nos permite ajustar la búsqueda del mínimo en su dirección opuesta.

$$W_{m+1} = W_m - \alpha F'(W_m) \quad (1)$$

Dentro del campo de la optimización de la gradiente se encuentran métodos como *Adadelta* (Severyn & Moschitti, 2015), *Adam* y *MSProp* (K, 2018), los cuales se utilizarán en la experimentación, pero antes es necesario conocerlos.

Adadelta es un método de aprendizaje por dimensión que se adapta dinámicamente con respecto al tiempo (Zeiler, 2012) y elimina la necesidad de experimentar con el factor de aprendizaje (Severyn & Moschitti, 2015). Este método restringe las ventanas de los gradientes anteriores que se acumulan con un tamaño fijo del peso y en su lugar realiza

una estimación local utilizando gradientes recientes, calculando así su promedio exponencialmente decreciente al cuadrado (Zeiler, 2012).

Adam es un método de optimización estocástica que requiere solo gradientes de primer orden, y a partir de ellos en un primer y segundo momento calcula la tasa de aprendizaje adaptativo para diferentes parámetros (Kingma & Ba, 2014).

MSPProp genera sus actualizaciones de parámetros usando una gradiente de re-escalado llamada *momentum* y puede dar pasos muy grandes en su operación, produciendo divergencia (Kingma & Ba, 2014).

1.4.2.2 Funciones de activación neuronal.

Como parte introductoria se explica brevemente cómo funciona la activación neuronal. La activación de cada neurona depende de algunos factores: i) las entradas (X_i), ii) pesos (W_i), iii) valor umbral (u) de activación, que es un índice sobre las unidades de las capas anteriores, y iv) una función de activación (f_a). Todo ello en conjunto da como resultado una salida X_j , cuyo proceso corresponde a la ecuación 2.

$$X_j = u + \sum_i F_a(X_i) * W_i \quad (2)$$

En la clasificación de varias clases, la última capa debe convertir sus entradas en una probabilidad, este proceso se logra implementando la función de activación *Sofmax*. Al usar *Sofmax* como función de salida el costo o pérdida a utilizarse es la entropía cruzada o *cross entropy*, pues esta pérdida relaciona las probabilidades con las salidas de la función y toma valores de uno o cero para determinar la clasificación (Hinton et al., 2012).

La función de activación determina un rango que definirá el estado de activación de una neurona (Barrera, 2018). Las funciones más comunes son sigmoideo, tangente hiperbólica (*tanh*) y rectificación lineal (*relu*) (Severyn & Moschitti, 2015), es por ello que se las utilizará en la experimentación. De estas funciones depende la velocidad de convergencia y la calidad de la solución obtenida (Severyn & Moschitti, 2015). En la tabla 1 se puede observar su función y el dominio, estas se utilizan para el cálculo de la gradiente y la activación de la red neuronal, por lo que se las detalla a continuación.

En la activación *sigmoid* el gráfico de su función se asemeja a una S, pues dentro de su dominio transcurre desde un nivel bajo acelerándose hasta llegar al clímax.

En la activación *relu también* llamada rectificadora, su gráfico se asemeja a una rampa, dentro de su dominio se mantiene en cero hasta dispararse al infinito.

En la activación *tanh* su función produce valores de dominio entre menos uno a uno, lo que nos indica que para su activación las variaciones que devuelve su función deben ser variaciones suaves entre los rangos positivos y negativos.

Tabla 1

Función de activación

Nombre	Función	Dominio
Sigmoid	$F(X_i) = \frac{1}{1 + e^{X_i}}$	(0,1)
Relu	$F(X_i) = \max(0, X_i)$	[0,∞)
Tanh	$F(X_i) = \frac{\sinh(X_i)}{\cosh(X_i)}$	(-1,1)

Esta tabla muestra las funciones de activación y sus dominios tomado de la fuente (Sharma, 2017).

1.4.2.3 Regularización

La regularización es un proceso que buscan evitar el sobreajuste, para ello, se han creado algunas técnicas como la disminución de pesos de las neuronas, *dropout* y *data augmentation* (Zhang, Bengio, Hardt, Recht, & Vinyals, 2016). De este conjunto, la elegida para implementarse en el modelo por su amplio uso es *Dropout* (Srivastava, Hinton, Krizhevsky, Sutskever, & Salakhutdinov, 2014).

Dropout es una técnica para quitar el sobreajuste mediante la eliminación de unidades neuronales, esta tarea se realiza a través de la supresión de conexiones durante el entrenamiento, lo cual evita que las unidades se adapten demasiado a los datos. Esta técnica ha mejorado el rendimiento en redes neuronales de aprendizaje supervisado en la clasificación de documento, sus valores típicos están en el rango de 0.5 a 0.8 (Srivastava et al., 2014). Su uso es implementado en conjunto de datos pequeños y medianos (Severyn & Moschitti, 2015).

1.4.2.4 Normalización por lotes (*BatchNormalization*)

Es una técnica que permite la fácil inicialización de pesos en la red neuronal, pues reduce la sensibilidad a los pesos iniciales. Tiene una activación de salida media de cero y una desviación estándar de uno, valores que le permiten estandarizar las entradas de las redes neuronales. Su objetivo es lograr una distribución estable de valores de activación durante el entrenamiento, es por eso que se aplican antes de la activación de capa evitando la distorsión en el tránsito de datos entre capas (Ioffe & Szegedy, 2015).

1.4.2.5 Épocas (*epochs*)

Una época es el proceso en el que el conjunto de datos completos ha transitado una vez por el modelo de la red neuronal hacia adelante y hacia atrás, este proceso actualiza los pesos de la gradiente descendente (Keras, n.d.).

1.4.2.6 Lotes (*batch_size*)

Un lote es la distribución de los datos de entrada en un conjunto de muestras que se procesan de manera paralela e independiente y que dan como resultado en conjunto una sola actualización del modelo (Keras, n.d.).

1.5 Evaluación de algoritmos

Una vez comprendidos los conceptos de los métodos de aprendizaje automático a evaluar, así como el aprendizaje profundo, su estructura y sus parámetros, es necesario determinar una medida de evaluar el rendimiento de los algoritmos.

Para la evaluación de algoritmos se cuenta con diversos métodos, los más comunes son: *precisión*, *recall*, *FMesasure*, *Acurracy* y *Error*. Sus nombres se encuentran en idioma inglés debido a que la traducción a nuestro idioma cambiaría su significado.

Las fórmulas que permiten el cálculo de estos métodos de evaluación se encuentran en las ecuaciones 3 – 7 y fueron tomadas de Wanjun & Xiaoguang (2010). El significado de la nomenclatura, teniendo en cuenta que se está clasificando elementos en una categoría X es el siguiente:

- i) TT son los positivos que fueron clasificados correctamente a la categoría.
- ii) NN son los negativos que fueron clasificados correctamente a la categoría.
- iii) TN son los positivos que fueron clasificados incorrectamente como negativos (falsos positivos).
- iv) NT son los negativos que fueron clasificados incorrectamente como positivos (falsos negativos).

$$Precision = \frac{TT}{TT+TN} \quad (3)$$

$$Recall = \frac{TT}{TT+NT} \quad (4)$$

$$FMeasure = \frac{2*Precision*Recall}{Precision+Recall} \quad (5)$$

$$Acurracy = \frac{TT+NN}{TT+TN+NT+NN} \quad (6)$$

$$Error = 1 - acurracy \quad (7)$$

Cada una de las ecuaciones 3 – 7 tienen una función en específico tomada de Junker, Hoch, & Dengel, (1999) que se detalla a continuación:

Precision tiene como objetivo encontrar la relación entre las instancias predichas correctamente como positivas y las instancias positivas totales. Determinando la cantidad de positivos que fueron clasificados incorrectamente como negativos.

Recall tiene como objetivo encontrar la relación entre las instancias predichas correctamente como positivas y las instancias totales. Determinando la cantidad de negativos que fueron clasificados incorrectamente como positivos.

FMeasure es una medida completa que se compone de *Precision* y *Recall*. Cuando su resultado es de uno el modelo es correcto y cuando es cero éste es inválido.

Acurracy tiene como objetivo indicar si el modelo está entrenando de manera adecuada y su funcionamiento es bueno en general.

Error es una medida que se compone del *accuracy* y muestra el margen en que el modelo funciona inadecuadamente.

De estos el método que se utiliza comúnmente para evaluar la categorización de texto es la exactitud o *accuracy* (Wanjun & Xiaoguang, 2010), la cuál será la métrica designada para la experimentación de los algoritmos.

2. MATERIALES Y MÉTODOS

2.1 Materiales

2.1.1 Datos

El conjunto de datos con el que se realizó la experimentación consta de 2,316 sílabos provenientes de la Universidad del Azuay y la Universidad de Cuenca divididos en nueve clases y 21 subclases detalladas en la tabla 2. Su clasificación fue realizada en base a las áreas y subáreas del conocimiento provistas por la UNESCO en su clasificación Internacional Normalizada de la Educación.

Debido a que se cuenta con un conjunto de datos de sílabos ya clasificados mencionados anteriormente, el tipo de aprendizaje que llevará acabo la red neural es supervisado. El aprendizaje supervisado es el método de entrenamiento más común en el aprendizaje automático. Este consiste en obtener datos etiquetados en categorías que durante el entrenamiento producen salidas en forma de vector de puntajes para cada clasificación, calculando una función objetivo que mide el error entre las salidas dadas y las deseadas (LeCun et al., 2015).

Tabla 2

Conjunto de sílabos preclasificados para el entrenamiento de los algoritmos

Áreas	Sub áreas	Cantidad
Humanidades y artes	Artes	386
Ciencias sociales educación comercial y derecho	Ciencias sociales y comportamiento	279
Ciencias sociales educación comercial y derecho	Educación comercial y administración	274
Programas generales	Desarrollo personal	174
Educación	Formación de personal docente y ciencias de educación	174
Ciencias	Matemática y estadística	157

Ciencias sociales educación comercial y derecho	Derecho	117
Ingeniería industria y construcción	Ingeniería y profesiones a fines	108
Humanidades y artes	Humanidades	107
Ingeniería industria y construcción	Industria y producción	97
Ciencias	Informática	76
Servicios	Servicios personales	65
Salud y servicios sociales	Medicina	64
Ciencias	Físicas	63
Ciencias	Ciencias de la vida	59
Ingeniería industria y construcción	Arquitectura y construcción	53
Agricultura	Agricultura silvicultura y pesca	29
Salud y servicios sociales	Servicios sociales	19
Servicios	Protección del medio ambiente	12
Programas generales	Programas de alfabetización y de aritmética	2
Programas generales	Programas básicos	1

Este conjunto de sílabos esta preclasificado tomando como criterios las áreas y subáreas dictaminadas por la UNESCO.

2.1.2 Software

El lenguaje de programación que se utilizó fue Python en su versión 3.6.5 instalado en un ambiente virtual *VirtualEnv*. Para la programación del modelo de aprendizaje profundo se utilizó la librería de redes neuronales *Keras* y para la evaluación de los algoritmos de aprendizaje automático, la librería de inteligencia artificial *Scikit-learn*. Los datos se encuentran centralizados en una base de datos *MySQL*.

2.1.3 Entorno de pruebas

La experimentación se realizó en un equipo con las siguientes características:

Modelo: MacBook Pro (Retina, 15-inch, Mid 2015)

Procesador: 2,2 GHz Intel Core i7

Memoria: 16 GB 1600 MHz DDR3

Gráficos: Intel Iris Pro 1536 MB

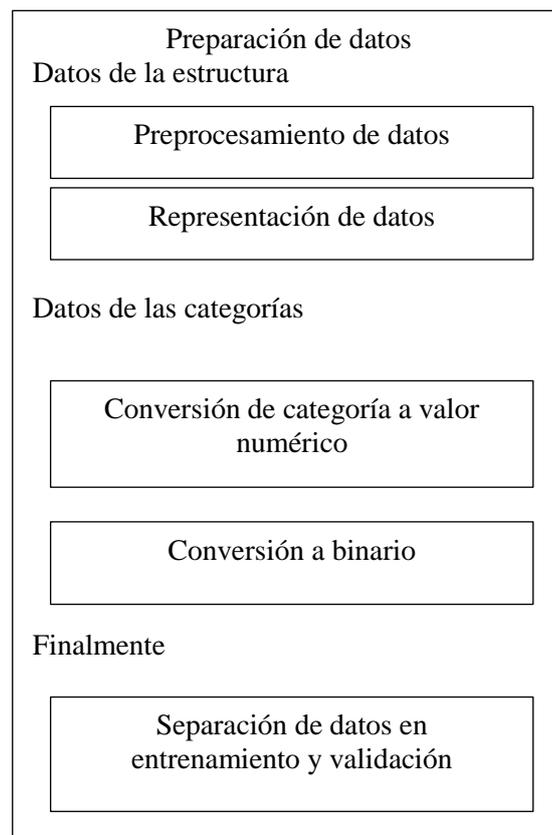
Sistema Operativo: macOS High Sierra Versión 10.13.5

2.2 Método

Este apartado consta de dos secuencias, el procesamiento de datos y la construcción del modelo computacional de aprendizaje profundo. Las ilustraciones 1 y 2 representan estas secuencias antes de ampliar su proceso en el documento.

Ilustración 1

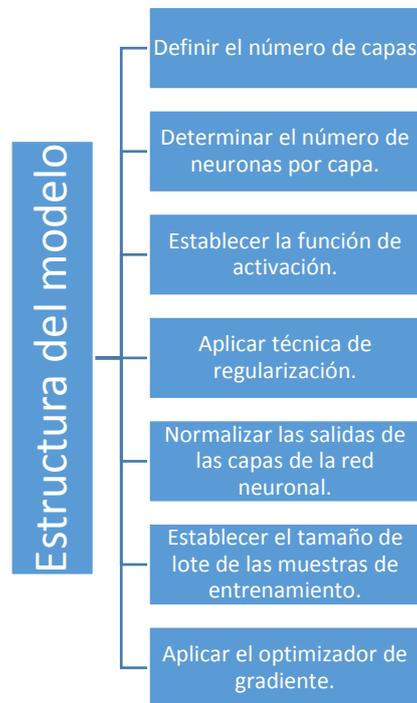
Procesamiento de datos



Este gráfico muestra el proceso que se realiza con los datos antes de introducirlos a la red neuronal.

Ilustración 2

Selección de elementos para estructurar el modelo



En esta estructuración de modelo corresponde al orden en el que se debe programar, pues primero se determina la estructura y luego la forma de ejecución.

2.2.1 Datos

Al conjunto de datos se realiza un **preprocesamiento** inicial, el cual es una tarea de suma importancia, pues la preparación de estos determina en gran medida la efectividad de un modelo. Dentro del proceso se busca eliminar o corregir datos anómalos y prepararlos para su ingreso a los algoritmos a evaluar (Kotsiantis et al., 2007).

El conjunto de pasos realizado en primera instancia en el preprocesamiento fue:

- Eliminar clases anómalas que no tenían un número de sílabos significativo para el entrenamiento y validación, estos datos corresponden al 2,72% del total.
- Transformar el texto a minúsculas, eliminar espacios, caracteres especiales y de puntuación mediante la librería *nlTK* de *Python*.
- Eliminar pronombres y conectores que no poseen valor semántico mediante la librería *Spacy* de *Python*.

Luego del preprocesamiento, el conjunto de datos debe representarse en información que una computadora pueda procesar, es decir en un modelo de representación (Wanjun & Xiaoguang, 2010). El método de **representación de datos** elegida es la denominada bolsa de palabras por su amplia utilización, el cual está basado en diccionarios. Cada bolsa de palabras contiene una matriz de la longitud total de elementos del diccionario, al ingresar un documento la matriz pondera con un valor de uno las palabras del diccionario que coinciden con las del documento y con un valor de cero las ausentes (Lai et al., 2015) (Joulin et al., 2016).

La clase que permitió realizar esta representación de datos es *CountVectorizer* de la librería *Scikit-learn* la cual cumple con el proceso de: i) *tokenizar* o dividir el texto en las palabras que lo componen ii) construir un vocabulario mediante el entrenamiento de los datos y iii) generar una matriz que relaciona los términos o palabras con el documento al que pertenece.

Luego de la creación de la bolsa de palabras, se facilita la inicialización de pesos en la red neuronal aplicando *StandarScaler*. Esta función cumple el mismo concepto de *BatchNormalization*, pero se utiliza solamente en la entrada de los datos hacia los algoritmos.

Una vez preparado el texto del contenido de los sílabos, el nombre de las categorías también debe cumplir con un proceso, este consiste en convertir el texto en un valor numérico y luego en un valor categórico binario al cual corresponderá la salida de la clasificación.

Una vez culminado los procesos previos es necesario **separar los datos** en un porcentaje adecuado para realizar dos tareas, el entrenamiento y la validación de los algoritmos. Para ello, en la experimentación se utilizó el 90% de los datos para entrenamiento y el 10% para la validación. Esta división fue realiza de manera aleatoria mediante la librería *Numpy* y sus valores fueron seleccionados tomando en cuenta el tamaño reducido del conjunto de datos. Pues si el conjunto de datos fuese más amplio el porcentaje de validación podría incrementarse.

2.2.2 Construcción del modelo computacional de aprendizaje profundo.

2.2.2.1 Estructura del modelo

La red de aprendizaje profundo a modelar es una red totalmente conectada, pues este tipo de redes son ampliamente utilizadas en diversas tareas de clasificación (K, 2018). Su entrenamiento se realiza de manera supervisada y consta de una estructura secuencial, es decir, es una pila lineal de capas densas.

Inicialmente la estructura con la que se empieza a experimentar fue basada en la referencia Peña (2017). Donde mediante una experimentación previa del número de neuronas y funciones de activación en un conjunto de datos de sentimiento de análisis, la red obtuvo un 78.66 % en entrenamiento y un 78.97 % en validación con un número de neuronas entre 128 – 256 por capa. La estructura inicial fue:

- a) Una capa con 256 neuronas.
- b) Normalización de lote de entrenamiento para su salida.
- c) Función de activación *relu*.
- d) Regularización *Dropout*.

Seguida de 2 estructuras con:

- a) Una capa con 128 neuronas.
- b) Normalización de lote de entrenamiento para su salida.
- c) Función de activación *relu*.
- d) Regularización *Dropout*.

Y como salida se coloca una capa con el número de clases a ser categorizadas y con una función *Softmax*. Todo este modelo se ejecuta mediante un tamaño de lote 64 muestras, con un optimizador de gradiente Adam.

Una vez definido el modelo, se tiene en cuenta que el número de capas cumple con el concepto antes mencionado en la revisión literaria de aprendizaje profundo y se calcula su exactitud mediante una validación cruzada de 10 iteraciones y un valor de época de cinco iteraciones, pues con este valor existe una relación estable entre el entrenamiento y la generalización.

2.2.2.2 Optimización de parámetros

Para determinar la combinación óptima de parámetros para el modelo computacional de aprendizaje profundo, se utiliza una cuadrícula de búsqueda. Los elementos que operarán dentro esta cuadrícula, llamada también grilla de parámetros, son los que se han seleccionado mediante la literatura y otros que se han expuesto en la tabla 3. Esta técnica de optimización permite una búsqueda simple de implementar y un resultado óptimo y confiable en espacios de baja dimensión como la categorización del texto (Bergstra & Bengio, 2012).

Tabla 3

Parámetros para la experimentación

Parámetro	Valores
Tamaño de lote de entrenamiento	16, 32, 64
Optimizador de gradiente	<i>RMSprop, Adadelta, Adam</i>
Número de neuronas	128, 256, 1024, 2048
Funciones de activación neuronal	<i>relu, tanh, sigmoid</i>

Esta tabla contiene los valores de los parámetros utilizados en la cuadrícula de búsqueda en la experimentación.

Debido a los recursos computacionales existentes y la falla en las ejecuciones por desborde de memoria, cuando se procesan muchos valores de parámetros, se opera mediante tres cuadrículas. Estas se ejecutan con valores de *Dropout* de 0.5 debido a que este número se lo considera como un valor por defecto (Srivastava et al., 2014) y época de entrenamiento de cinco iteraciones, pues con este valor existe una relación estable entre el entrenamiento y la generalización.

La cuadrícula se ejecuta mediante la clase *GridSearchCV* de la librería *Scikit-learn*, la cual da como resultado las combinaciones de parámetros por orden de puntaje. Este puntaje calcula la exactitud mediante una validación cruzada de tres instancias.

La primera cuadrícula ejecutada contenía los siguientes parámetros:

- a) Tamaño de lote de entrenamiento: 16, 32, 64.
- b) Optimizador de gradiente: *RMSprop, Adadelta, Adam*.
- c) Número de neuronas en las capas: 128, 256, 1024.
- d) Funciones de activación neuronal: *relu, tanh, sigmoid*.

Luego de la ejecución se obtuvo cuatro modelos que se destacan por su alto puntaje, con rangos que oscilan entre 68.85% y 68.03%. Los valores de las combinaciones paramétricas que no se encuentran en las puntuaciones altas de la cuadrícula se suprimen. Estos valores son el tamaño de lote de entrenamiento de 64 muestras, el optimizador *Adadelta*, el número de neuronas de 128, 256 y la función de activación *relu*.

Luego del descarte de elementos, se añaden valores inferiores y superiores en el rango de los parámetros que en su ejecución mostraron tener los mejores resultados. Por ejemplo, dentro del rango de tamaño por lote de entrenamiento, dos modelos tenían como valor límite inferior 16, por lo que es necesario comprobar si un valor límite más bajo constituye una mejora en el modelo dado. Es así como adicionando los valores en base al criterio ejemplificado tenemos la segunda cuadrícula:

- a) Tamaño de lote de entrenamiento: 8, 16, 32.
- b) Optimizador de gradiente: *RMSprop*, *Adam*.
- c) Número de neuronas en las capas: 512, 1024, 2048.
- d) Funciones de activación neuronal: *tanh*, *sigmoid*.

En la ejecución se obtiene dos modelos con altos puntajes, uno estable debido a que sus valores se mantienen entre los rangos ya definidos y otro que se tiene que adicionar valores en sus límites para comprobar su estabilidad. Para ello se estructuró la siguiente cuadrícula:

- e) Tamaño de lote de entrenamiento: 4, 8, 16.
- f) Optimizador de gradiente: *RMSprop*.
- g) Número de neuronas en las capas: 1024, 2048.
- h) Funciones de activación neuronal: *tanh*.

Finalmente se obtiene un modelo con una combinación óptima de parámetros. Como aspecto experimental se aumentó el número de capas de la red neuronal, hecho que disminuyó la exactitud. Estos resultados desfavorables en el aumento de capas indicaron que el modelo optimizado resultante del primer lugar en la última ejecución de la cuadrícula es el correcto. Y éste es el que será comparado mediante la medida de exactitud con los algoritmos de aprendizaje automático ya seleccionados.

2.2.2.3 Evaluación de algoritmos

Los algoritmos a evaluar son el aprendizaje profundo, los basados en reglas de decisión, aprendizaje estadístico y *SVM*. La medida de exactitud, que será la forma de evaluación de los algoritmos, se calculará en base a una validación cruzada de 10 instancias. El número de instancias se escogió de manera experimental y tomando en cuenta los recursos computacionales.

Para la evaluación de la categoría de árboles de decisión se utiliza la clase *Random Forest Classifier* que provee un conjunto de algoritmos ensamblados que generan árboles de clasificación. En el caso de la categoría de aprendizaje estadístico se implementa mediante la clase *Naive Bayes Multinomial* y con respecto a *SVM* se implementa con la clase *linearSVM* pues esta provee un algoritmo con una gran flexibilidad en la elección de penalización de pesos y es totalmente escalable a mayor cantidad de muestras.

2.2.2.4 Repetición de datos.

Luego de obtener la exactitud del modelo de aprendizaje profundo, así como de los algoritmos de *SVM*, árboles de decisión y *Naive Bayes*, experimentalmente se realizó el aumento de contenido en el conjunto de datos mediante la repetición de elementos. El porcentaje de repetición varía en el rango de cero al 20 por ciento y su selección de elementos es aleatoria.

Esta tarea tiene el objetivo de encontrar el comportamiento de los algoritmos antes mencionados con respecto a la repetición porcentual de elementos al conjunto de datos. Se espera que con este proceso el modelo computacional de aprendizaje profundo se estabilice y supere por mucho a los algoritmos comparados.

2.2.2.5 La exactitud como medida de evaluación de algoritmos

El artículo de Li, Zhang, & Ogihara (2004) menciona que en categorías con muchas clases en conjuntos de datos imbalancados, la exactitud como valor se degrada y recomienda el uso de otras medidas para el proceso. Con el fin de validar la exactitud como medida para la experimentación realizada en esta investigación se usa la función de la clase *class_weight*, tanto en el aprendizaje profundo como en los algoritmos que fueron evaluados. Esta clase balancea el conjunto de datos de entrenamiento y permitirá ver el resultado de la generalización en el proceso.

3. RESULTADOS

3.1 Construcción del modelo computacional de aprendizaje profundo

3.1.1 Estructura del modelo

Una vez determinada la estructura del modelo computacional de aprendizaje profundo se pasa a evaluarlo. Como resultado se muestran dos valores de exactitud, el primer valor corresponde al entrenamiento de la red de aprendizaje profundo y el segundo valor hace referencia a la validación de un conjunto de datos nuevos en la red.

En primera instancia el resultado de la evaluación fue de 93.63% en entrenamiento y un 61.08% en pruebas, estos valores indican que en la red de aprendizaje profundo existe un sobreajuste. Esto significa que la red está aprendiendo de manera más ajustada los datos que por ella transitan, provocando una incorrecta generalización, pues existe una diferencia de 32.55% entre la exactitud de entrenamiento y la de validación. Teniendo en cuenta los recursos computacionales y el tamaño del conjunto de datos se busca reducir el margen de sobreajuste mediante una optimización de parámetros.

3.1.2 Optimización de parámetros

La primera cuadrícula ejecutada dio como resultados las combinaciones de la tabla 4, donde podemos observar que el puntaje asignado a las mejores combinaciones oscila dentro del 68%.

Tabla 4

Ejecución de la primera cuadrilla.

Posición	Puntaje %	Función de Activación	Tamaño del lote	Abandono %	Época	Número de neuronas	Optimizador
1	68.85	sigmoid	32	0.5	5	1024	RMSprop
2	68.55	sigmoid	16	0.5	5	1024	Adam
3	68.29	tanh	16	0.5	5	1024	RMSprop
4	68.03	sigmoid	32	0.5	5	1024	Adam

Esta tabla contiene la combinación de parámetros resultado de la ejecución de la primera cuadrícula.

Luego de preparar la segunda cuadrícula descartando elementos que no se encontraron dentro de las mejores combinaciones y aumentando valores en los parámetros en base al análisis de los primeros resultados, se obtuvieron las combinaciones expuestas en la tabla 5.

Se observa que la combinación que lleva el segundo lugar se estabilizó, pues sus valores se encuentran dentro de los rangos de sus parámetros, esto hace necesario experimentar con el cambio de los valores límites de la combinación que lleva el primer lugar, con el fin de determinar si este cambio genera alguna alteración en la combinación de mejores parámetros.

Tabla 5

Ejecución de la segunda cuadrícula

Posición	Puntaje %	Función de Activación	Tamaño del lote	Abandono %	Época	Número de neuronas	Optimizador
1	68.81	tanh	8	0.5	5	2048	RMSprop
2	68.77	sigmoid	32	0.5	5	1024	RMSprop
3	68.46	tanh	8	0.5	5	1024	RMSprop
4	68.25	sigmoid	16	0.5	5	2048	RMSprop

Esta tabla contiene la combinación de parámetros resultado de la ejecución de la segunda cuadrícula.

Una vez que se realizó la ejecución de la tercera cuadrícula, se observa en la tabla 6 que el puntaje mayor corresponde a la combinación de parámetros que obtuvo el lugar número uno en la tabla 5. Lo que muestra que la primera combinación de la tabla 5 es el resultado con la combinación óptima de parámetros para la experimentación.

Tabla 6

Ejecución de la tercera cuadrícula

Posición	Puntaje %	Función de Activación	Tamaño del lote	Abandono %	Época	Número de neuronas	Optimizador
1	69.67	tanh	8	0.5	5	2048	RMSprop
2	68.16	tanh	16	0.5	5	2048	RMSprop
3	68.16	tanh	4	0.5	5	4096	RMSprop

Esta tabla contiene la combinación de parámetros resultado de la ejecución de la tercera cuadrícula, la cual muestra la combinación óptima de parámetros encontrada.

La ejecución del modelo con una validación cruzada de 10 iteraciones da como resultado una exactitud de 80.58% en entrenamiento y en validación 70.44%.

El primer aspecto que se experimentó con el modelo ya estructurado fue el aumento de número de capas y las épocas de entrenamiento a 100 iteraciones. El resultado mostró en conclusión que a mayor número de capas se establece un balance entre los valores de exactitud en entrenamiento y validación en un inicio, perdiéndose este a medida que aumentan las épocas. Además, el modelo alcanza una exactitud de validación inferior a la estructura ya establecida como óptima. La tabla 7 muestra el aumento de capas, la época con el mejor resultado y sus valores de exactitud.

Tabla 7

Aumento de número de capas.

Número de capas	Exactitud de entrenamiento	Exactitud de validación	Época
9	91.56 %	66.36 %	97
11	87.08 %	57.14%	93
13	87.43 %	57.64 %	13

Esta tabla muestra la exactitud en entrenamiento y validación con respecto al número de capas y su mejor época.

3.1.3 Evaluación de algoritmos

Una vez establecido el modelo de aprendizaje profundo junto con los parámetros optimizados procedemos a evaluarlo en contraste con los algoritmos ya seleccionados, dando como resultado la tabla 8. El cálculo de esta exactitud se la realiza con una validación cruzada de 10 iteraciones de las cuales el primer lugar en la clasificación de sílabos obtuvo el aprendizaje profundo, seguido del algoritmo *Naive Bayes*, *SVM* y el árbol de decisión.

Tabla 8

Tabla comparativa de algoritmos en la clasificación de sílabos.

Número	Algoritmo	Exactitud de validación
1	Aprendizaje Profundo	70.44 %
2	Naive Bayes	69.01 %
3	SVM	64.21 %
4	Árbol de decisión	63.22 %

Esta tabla contiene la exactitud de cada algoritmo en la tarea de clasificación de sílabos en las áreas y subáreas dictaminadas por la UNESCO. Su orden se encuentra de acuerdo a la exactitud en forma descendente.

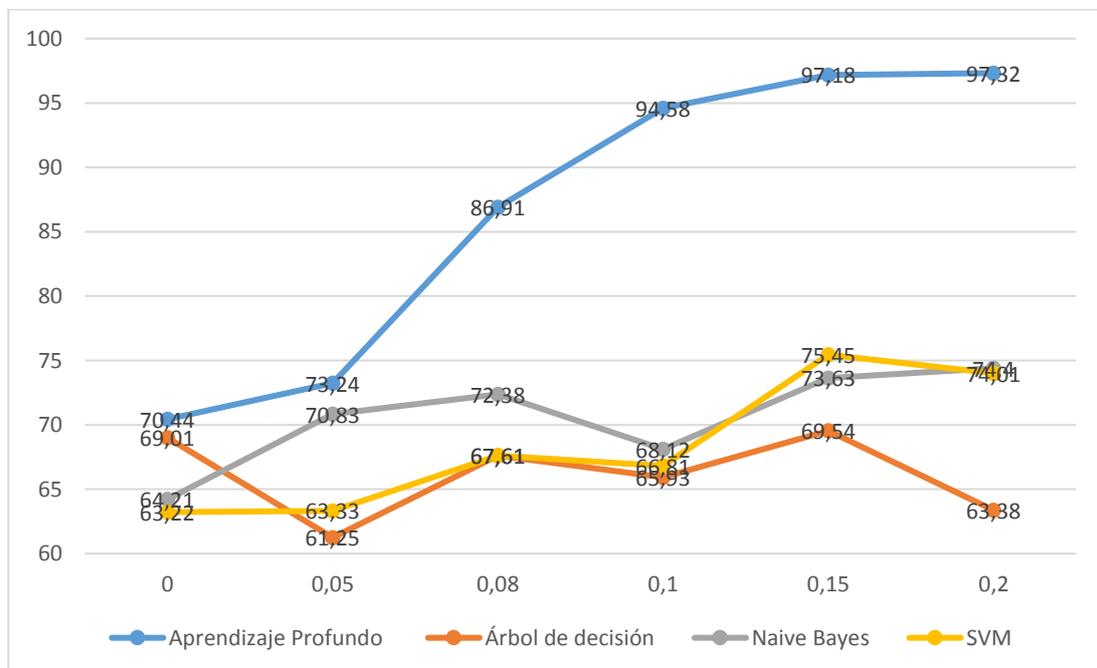
3.1.4 Repetición de datos

Luego de aumentar el contenido del conjunto de datos porcentualmente, se puede ver el aumento exponencial en la exactitud de validación del aprendizaje profundo frente al comportamiento oscilante de los demás algoritmos, la ilustración 3 muestra este hecho.

Cabe destacar que este crecimiento exponencial en la exactitud de validación debe guardar relación con el crecimiento de la exactitud de entrenamiento para validar un modelo sin mucho porcentaje de sobreajuste.

Ilustración 3

Gráfico de aumento de datos.



Comparación de algoritmos con la repetición de datos.

3.1.5 La exactitud como medida de evaluación de algoritmos.

En el caso de este proyecto a pesar de tener un conjunto de datos no balanceados, la proporción entre número de elementos de las clases y su dimensionalidad no distorsiona ni degrada la exactitud. La proporción entre la clase con mayor número de sílabos y la menor es de 1:8 luego del procesamiento de datos. Los resultados de la experimentación en el aprendizaje profundo fueron:

Con clases balanceadas: 78.03% en entrenamiento y un 68.35% de validación.

Con clases no balanceadas: 80.58% en entrenamiento y un 70.44% de validación.

Cabe destacar que el margen entre la exactitud de entrenamiento - validación, tanto en las clases balanceadas e imbalanceadas, tiene una media de 9.91% con una desviación de 0.22. Su diferencia de porcentajes en sus resultados se debe a la aleatoriedad de elementos que se toman para el entrenamiento y validación de la red de aprendizaje profundo.

En la evaluación del aprendizaje profundo frente a los algoritmos *SVM*, *Naive Bayes* y Árboles de decisión tenemos como resultado que los puntajes de exactitud corresponden al mismo orden tanto con clases balanceadas como imbalanceadas, tal como lo describe la tabla 9. Es por eso que como conclusión se determina que no existe un factor que evidencie un cambio sustancial y anómalo para desechar esta medida o complementarla con otra forma de evaluación de algoritmos.

Tabla 9

Comparación de algoritmos con clases balanceadas y no balanceadas.

Número	Algoritmo	Exactitud de validación sin balancear entrenamiento	Exactitud de validación balanceado el entrenamiento
1	Aprendizaje Profundo	70.44 %	69.45%
2	Naive Bayes	69.01 %	69.41%
3	SVM	64.21 %	62.95%
4	Árbol de decisión	63.22 %	61.03%

Esta tabla muestra la exactitud de los algoritmos seleccionados, con clases balanceadas y no balanceadas en su entrenamiento.

4. DISCUSIÓN Y CONCLUSIONES

En este artículo se estructura un modelo de aprendizaje profundo con el fin de clasificar planes de estudios académicos. Una vez concluida la construcción del modelo computacional y su optimización de parámetros se realiza una comparación entre el aprendizaje profundo y algoritmos basados en *SVM*, *Naive Bayes* y Árboles de decisión.

La experimentación muestra que el modelo generado en primera instancia como punto de partida tuvo una diferencia de exactitud entre entrenamiento y validación de 32.55% con efecto de sobreajuste. Esto se da frecuentemente cuando se trabaja con un conjunto de datos pequeños y medianos (Severyn & Moschitti, 2015). Con la finalidad de reducir el problema se optimizaron los parámetros y se aplicó un método de regularización (Zhang et al., 2016). Como consecuencia del proceso mencionado el margen de diferencia entre la exactitud de entrenamiento y validación es de 10.14%, margen aceptable, que muestra una mejor generalización, es decir, menos sobreajuste. Se espera que con el aumento del conjunto de datos este margen disminuya más y aumente la exactitud del modelo en entrenamiento y validación.

En la comparación entre el modelo de aprendizaje profundo y los algoritmos ya mencionados que se encuentra en la tabla 8, se obtiene que el aprendizaje profundo optimizado tiene una exactitud ligeramente superior a los demás algoritmos que se evaluaron con sus valores por defecto. Esto se debe a que el aprendizaje profundo funciona mejor descubriendo estructuras en grandes conjuntos de datos (LeCun et al., 2015). A pesar de los datos limitados la superioridad porcentual del aprendizaje profundo es una buena opción para la clasificación de sílabos.

Este estudio concentró un acercamiento hacia el aprendizaje profundo en la clasificación de sílabos con una estructura y un conjunto de parámetros optimizados por una cuadrícula, cuya ejecución requiere recursos computacionales altos, por lo tanto, para experimentar fue necesario determinar un conjunto de valores de parámetros en concordancia con la literatura revisada. Se propone en futuros trabajos utilizar métodos meta heurísticos con el fin de optimizar y evaluar una mayor cantidad de parámetros con los recursos computacionales accesibles al investigador. A más de ello se propone el uso de *data augmentation* o aumento de data como técnica para nivelar y aumentar la exactitud del aprendizaje profundo en su entrenamiento y validación. Esta técnica

elegante se basa en deformaciones de una colección de muestras de entrenamiento para adicionarlas en el conjunto original (Salamon & Bello, 2017). Con el fin de mostrar el comportamiento de una cantidad mayor de datos en los algoritmos se realizó la experimentación, que da como resultado lo descrito en la ilustración 3, donde el aprendizaje profundo tiene un crecimiento de validación exponencial a la cantidad de datos que por ella se entrenen.

En conclusión, el modelo de red generado puede ser utilizado como clasificador de sílabos en áreas y subáreas de la UNESCO, con el conjunto de datos de entrenamiento de 70.44 % de exactitud en validación. A pesar de que la exactitud podría mejorarse; el resultado obtenido con el conjunto de datos existentes es óptimo, ya que la técnica de aprendizaje profundo fue superior a los algoritmos comparados expuestos en la tabla 8.

Se espera que el modelo computacional construido en este artículo se considere como modelo base para aplicar un aumento de contenido al conjunto de datos y así lograr una mayor exactitud en entrenamiento - validación y un menor sobreajuste. Al implementarlo en su contexto operacional permitirá ahorrar el arduo trabajo del personal académico en la clasificación de sílabos en áreas y subáreas de la UNESCO.

BIBLIOGRAFÍA

- Abdiansah, A., & Wardoyo, R. (2015). Time complexity analysis of support vector machines (SVM) in LibSVM. *International Journal Computer and Application*.
- Arlot, S., Celisse, A., & others. (2010). A survey of cross-validation procedures for model selection. *Statistics Surveys*, 4, 40–79.
- Barrera, J. A.-T. (20 de 05 de 2018). Redes Neuronales. Obtenido de http://www.cucei.udg.mx/sites/default/files/pdf/toral_barrera_jamie_areli.pdf
- Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb), 281–305.
- Bowker, G. C., & Star, S. L. (2000). *Sorting things out: Classification and its consequences*. MIT press.
- Chung, H., & Kim, J. (2016). An ontological approach for semantic modeling of curriculum and syllabus in higher education. *International Journal of Information and Education Technology*, 6(5), 365.
- Deng, L., & Yu, D. (2014). Deep learning: methods and applications. *Foundations and Trends® in Signal Processing*, 7(3–4), 197-387
- der Aalst, W. M. P., Rubin, V., Verbeek, H. M. W., van Dongen, B. F., Kindler, E., & Günther, C. W. (2010). Process mining: a two-step approach to balance between underfitting and overfitting. *Software & Systems Modeling*, 9(1), 87.
- Doolittle, P. E., & Lusk, D. L. (2007). The Effects of Institutional Classification and Gender on Faculty Inclusion of Syllabus Components. *Journal of the Scholarship of Teaching and Learning*, 7(2), 62–78.
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A., Jaitly, N., ... others. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine*, 29(6), 82–97.
- Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *ArXiv Preprint ArXiv:1502.03167*.
- Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *ArXiv Preprint ArXiv:1607.01759*.
- Johnson, R., & Zhang, T. (2014). Effective use of word order for text categorization with convolutional neural networks. *arXiv preprint arXiv:1412.1058*.
- Junker, M., Hoch, R., & Dengel, A. (1999). On the evaluation of document analysis components by recall, precision, and accuracy. In *Document Analysis and Recognition, 1999. ICDAR'99. Proceedings of the Fifth International Conference on* (pp. 713–716).
- Keras. (s.f.). Obtenido de The Python Deep Learning Library: <https://keras.io/>
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *ArXiv Preprint ArXiv:1412.6980*.
- K, L. (18 de Enero de 2018). RapidMiner. Obtenido de An Introduction to Deep Learning with RapidMiner: <https://rapidminer.com/resource/state-deep-learning/>
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160, 3–24.
- Lai, S., Xu, L., Liu, K., & Zhao, J. (2015). Recurrent Convolutional Neural Networks for Text Classification. In *AAAI* (Vol. 333, pp. 2267–2273).

- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.
- Li, T., Zhang, C., & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15), 2429–2437.
- Ongsulee, P. (2017). Artificial intelligence, machine learning and deep learning. In *ICT and Knowledge Engineering (ICT&KE), 2017 15th International Conference on* (pp. 1–6).
- Ota, S., & Mima, H. (2011). Machine learning-based syllabus classification toward automatic organization of issue-oriented interdisciplinary curricula. *Procedia-Social and Behavioral Sciences*, 27, 241-247.
- Peña, V. (2 de 9 de 2017). GitHub. Obtenido de Text Classification with Keras and TensorFlow: <https://gist.github.com/vgpena/b1c088f3c8b8c2c65dd8edbe0eae7023/revisions>
- RAE. (2017). clasificacion. Obtenido de <http://dle.rae.es/?id=9PqQrQv>
- Salamon, J., & Bello, J. P. (2017). Deep convolutional neural networks and data augmentation for environmental sound classification. *IEEE Signal Processing Letters*, 24(3), 279–283.
- Severyn, A., & Moschitti, A. (2015). Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 373–382).
- Sharma, S. (6 de Septiembre de 2017). Obtenido de Activation Functions: Neural Networks: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1), 1929–1958.
- Tang, B., He, H., Baggenstoss, P. M., & Kay, S. (2016). A Bayesian classification approach using class-specific features for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, 28(6), 1602–1606.
- Wanjuan, Y., & Xiaoguang, S. (2010). Research on text categorization based on machine learning. In *Advanced Management Science (ICAMS), 2010 IEEE International Conference on* (Vol. 2, pp. 253–255).
- Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... others. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1), 1–37.
- Yu, X., Tungare, M., Fan, W., Perez-Quinones, M., Fox, E. A., Cameron, W., ... Cassel, L. (2007). Automatic syllabus classification. In *Proceedings of the 7th ACM/IEEE-CS joint conference on Digital libraries* (pp. 440–441).
- Zaremba, W., Sutskever, I., & Vinyals, O. (2014). Recurrent neural network regularization. *ArXiv Preprint ArXiv:1409.2329*.
- Zeiler, M. D. (2012). ADADELTA: an adaptive learning rate method. *ArXiv Preprint ArXiv:1212.5701*.
- Zhang, C., Bengio, S., Hardt, M., Recht, B., & Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *ArXiv Preprint ArXiv:1611.03530*.

Doctora María Elena Ramírez Aguilar, Secretaria de la Facultad de Ciencias de la Administración de la Universidad del Azuay

CERTIFICA:

Que, el Consejo de Facultad en sesión del 11 de mayo de 2018, conoció y aprobó la solicitud para realización del trabajo de titulación, presentada por :

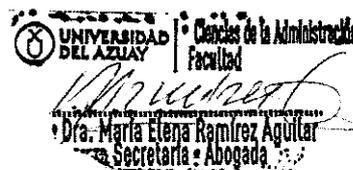
Estudiante: Juan Marcelo Alvarado Carrera (cód. 67072)
Tema: “Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo”
Previo a la obtención del título de Ingeniero de Sistemas y Telemática
Director: Ing. Marcos Orellana ✓
Tribunal: Ing. Gerardo Orellana ✓
Ing. Fernando Balarezo ✓

Plazo de presentación del trabajo de titulación, con la calificación del Director: seis meses a partir de la fecha de aprobación, esto es hasta el 11 de noviembre de 2018.

E INFORMA:

Que en aplicación de la Disposición General Cuarta del Reglamento de Régimen Académico vigente, en caso de que el estudiante no culmine y apruebe el trabajo de titulación luego de dos períodos académicos contados a partir de la fecha de culminación de estudios, deberá realizar la actualización de conocimientos previa a su titulación.

Cuenca, 14 de mayo de 2018



UNIVERSIDAD DEL AZUAY | Ciencias de la Administración
Facultad
Dra. María Elena Ramírez Aguilar
Secretaria Abogada

CONVOCATORIA

Por disposición de la Junta Académica de la escuela de Ingeniería de Sistemas y Telemática se convoca a los Miembros del Tribunal Examinador, a la sustentación del Protocolo del Trabajo de Titulación: "**Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo**", presentado por el estudiante Juan Marcelo Alvarado Carrera con código 67072; previa a la obtención del título de Ingeniero de Sistemas y Telemática, para el día **Miércoles, 09 de mayo de 2018 a las 08:00.**

Tomar en cuenta que posterior a la sustentación del Diseño del Trabajo de Titulación, por ningún concepto se puede realizar modificaciones ni cambios en los documentos; únicamente, en caso de diseño aprobado con modificación, el Director adjuntará al esquema un oficio indicando que se procede con los cambios sugeridos.

Cuenca, 07 de mayo de 2018



Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad

Ing. Marcos Orellana



Ing. Gerardo Orellana



Ing. Fernando Balarezo



ESCUELA DE INGENIERÍA DE SISTEMAS Y TELEMÁTICA

FECHA: 07 de MAYO DE 2018.

Estudiante: Juan Marcelo Alvarado Carrera

Oficio Nro. 018-2018-DIST-UDA

Cuenca, 2 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 2 de mayo del 2018, recibió el proyecto de tesis titulado "Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo", presentado por Juan Marcelo Alvarado Carrera estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Marcos Orellana, como co-director y miembro de tribunal al Ing. Gerardo Orellana, y como miembro de tribunal al Ing. Fernando Balarezo.

Atentamente,



Ing. Catalina Astudillo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay



Ing. Esteban Crespo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay



ACTA
SUSTENTACIÓN DE PROTOCOLO/DENUNCIA DEL TRABAJO DE TITULACIÓN

Fecha de sustentación: **Miércoles, 09 de mayo de 2018 a las 08:00**

- 1.1. Nombre del estudiante: Juan Marcelo Alvarado Carrera
1.2. Código: 67072
1.3. Director sugerido: Ing. Marcos Orellana
1.4. Codirector (opcional): _____
1.4.1. Tribunal: Ing. Gerardo Orellana e Ing. Fernando Balarezo
1.4.2. Título propuesto: **“Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo”**
1.4.3. Aceptado sin modificaciones :

1.4.4. Aceptado con las siguientes modificaciones:

1.4.5. No aceptado

1.4.6. Justificación:

Ing. Marcos Orellana

Tribunal

Ing. Gerardo Orellana

Ing. Fernando Balarezo

Sr. Juan Marcelo Alvarado Carrera

Dra. María Elena Ramírez Aguilar
Secretaria de la Facultad



RÚBRICA PARA LA EVALUACIÓN DEL PROTOCOLO DE TRABAJO DE TITULACIÓN
(Tribunal)

- 1.1. Nombre del estudiante: Juan Marcelo Alvarado Carrera
1.2. Código : 67072
1.3. Director sugerido:
1.3.1. Codirector (opcional): Ing. Marcos Orellana
1.3.2. Título propuesto: **“Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo”**
1.4. Revisores tribunal: Ing. Gerardo Orellana e Ing. Fernando Balarezo
1.5. Recomendaciones generales de la revisión:

	Cumple	No cumple
Problemática y/o pregunta de investigación		
1. ¿Presenta una descripción precisa y clara?	✓	
2. ¿Tiene relevancia profesional y social?	✓	
Objetivo general		
3. ¿Concuerda con el problema formulado?	✓	
4. ¿Se encuentra redactado en tiempo verbal infinitivo?	✓	
Objetivos específicos		
5. ¿Permiten cumplir con el objetivo general?	✓	
6. ¿Son comprobables cualitativa o cuantitativamente?	✓	
Metodología		
7. ¿Se encuentran disponibles los datos y materiales mencionados?	✓	
8. ¿Las actividades se presentan siguiendo una secuencia lógica?	✓	
9. ¿Las actividades permitirán la consecución de los objetivos específicos planteados?	✓	
10. ¿Las técnicas planteadas están de acuerdo con el tipo de investigación?		
Resultados esperados		
11. ¿Son relevantes para resolver o contribuir con el problema formulado?	✓	
12. ¿Concuerdan con los objetivos específicos?	✓	
13. ¿Se detalla la forma de presentación de los resultados?		
14. ¿Los resultados esperados son consecuencia, en todos los casos, de las actividades mencionadas?	✓	

Ing. Marcos Orellana

Ing. Gerardo Orellana

Ing. Fernando Balarezo

Oficio Nro. 018-2018-DIST-UDA

Cuenca, 2 de mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De nuestras consideraciones:

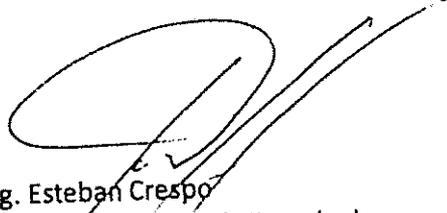
La Junta Académica de la Escuela de Ingeniería de Sistemas y Telemática, reunida el día 2 de mayo del 2018, recibió el proyecto de tesis titulado "Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo", presentado por Juan Marcelo Alvarado Carrera estudiante de la Escuela de Ingeniería de Sistemas y Telemática, y revisado por el Ing. Marcos Orellana previo a la obtención del título de Ingeniero de Sistemas y Telemática.

Por lo expuesto, y de conformidad con el Reglamento de Graduación de la Facultad, recomendamos como director y responsable de aplicar cualquier modificación al diseño del trabajo de graduación posterior al Ing. Marcos Orellana, como co-director y miembro de tribunal al Ing. Gerardo Orellana, y como miembro de tribunal al Ing. Fernando Balarezo.

Atentamente,



Ing. Catalina Astudillo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay



Ing. Esteban Crespo
Junta Académica de la Escuela de
Ingeniería de Sistemas y Telemática
Universidad del Azuay

Cuenca, 4 de Mayo del 2018

Ingeniero,

Oswaldo Merchán Manzano

DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACION

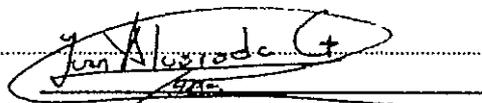
UNIVERSIDAD DEL AZUAY

De mi consideración,

Estimado Señor Decano, yo **Juan Marcelo Alvarado Carrera** con C.I. **0106964000**, código estudiantil **67072** estudiante de la carrera de Ingeniería de Sistemas y Telemática, solicito muy comedidamente a usted y por su intermedio al Consejo de Facultad, la aprobación del protocolo de trabajo de titulación con el tema: **"Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo"** previo a la obtención del título de Ingeniero en Sistemas y Telemática para lo cual adjunto la documentación respectiva.

Por la favorable acogida que brinde a la presente, anticipo mis agradecimientos.

Atentamente



Juan Marcelo Alvarado Carrera

Estudiantes de la Carrera de Sistemas y Telemática

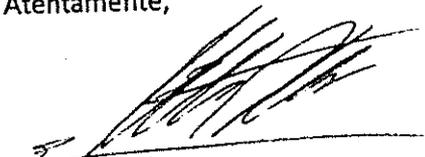
Cuenca, 4 de Mayo de 2018

Señor Ingeniero
Oswaldo Merchán Manzano
DECANO DE LA FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN
Presente.-

De mis consideraciones:

Luego de revisar el diseño del trabajo de graduación denominado "Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo", presentado por Juan Marcelo Alvarado Carrera estudiante de la Escuela de Ingeniería de Sistemas y Telemática. Considero que el documento cumple con las normas legales y reglamentarias de la Universidad y de la Facultad de Ciencias de la Administración, por lo que recomiendo su aprobación por parte del Consejo de Facultad.

Atentamente,



Ing. Marcos Orellana Cordero
Director



UNIVERSIDAD DEL
AZUAY

DOCTORA MARÍA ELENA RAMÍREZ AGUILAR, SECRETARIA DE LA
FACULTAD DE CIENCIAS DE LA ADMINISTRACIÓN DE LA UNIVERSIDAD DEL
AZUAY

CERTIFICA:

Que, el señor **ALVARADO CARRERA JUAN MARCELO** con código **67072**, alumno
de la carrera de **INGENIERIA DE SISTEMAS Y TELEMATICA**, tiene aprobado el
89% de créditos de su malla curricular.

Cuenca, 26 de abril de 2018

Dra. María Elena Ramírez Aguilar
**SECRETARIA DE LA FACULTAD
DE CIENCIAS DE LA ADMINISTRACIÓN**



UNIVERSIDAD DEL
AZUAY
FACULTAD DE
ADMINISTRACION
SECRETARIA

Derecho No.: 001-001-000171802
mjmr.-



UNIVERSIDAD DEL
AZUAY

UNIVERSIDAD DEL AZUAY
INGENIERÍA EN SISTEMAS Y TELEMÁTICA
DISEÑO DE TESIS

1. DATOS GENERALES

1.1 Nombre del estudiante: Juan Marcelo Alvarado Carrera

1.1.1 Código: 067072

1.1.2 Contacto

1.1.2.1 Celular: 0958727629

1.1.2.2 Correo: jmarceloalvarado94@hotmail.com

1.2 Director sugerido: Ing. Marcos Orellana Cordero

1.2.1 Contacto

1.2.1.1 Celular: 0999955611

1.2.1.1 Correo: marore@uazuay.edu.ec

1.3 Co-director sugerido: Msc. Gerardo Orellana

1.3.1 Contacto

1.3.1.1 Celular: 0998163786

1.3.1.2 Correo: gorellana@uazuay.edu.ec

1.4 Asesor metodológico:

1.5 Tribunal designado:

1.6 Aprobación:

1.7 Línea de Investigación de la carrera:

1.7.1 Código UNESCO: 1203 Ciencias de los ordenadores

1203. 04 Inteligencia artificial

1.7.2 Tipo de trabajo: Investigación

1.8 Área de estudio: Redes neuronales

1.9 Título propuesto: Clasificación de sílabos académicos en base a redes neuronales de aprendizaje profundo.

1.10. Subtítulo:

1.11 Estado del proyecto:

En el grupo de investigación LIDI, existe un proyecto en curso denominado "Detección de similitudes entre contenidos académicos de carrera a través de la aplicación de tecnologías semánticas y minería de datos" como parte del concurso CEPRA IX, que tiene como objetivo homologar créditos académicos entre instituciones de educación superior, verificando la similitud de sus sílabos. Para este proyecto del grupo de investigación, se utiliza Support Vector Machine (SVM), que es un algoritmo de aprendizaje supervisado, para clasificar cada sílabo en las áreas y sub áreas de conocimiento propuestas por la UNESCO.

El proyecto que se pretende realizar tiene el fin de clasificar los planes académicos mediante redes neuronales de aprendizaje profundo, además de comparar esta técnica con varios algoritmos para clasificación de textos, entre ellas la utilizada actualmente en la investigación descrita al inicio.

2. CONTENIDO

2.1 Motivación de la investigación:

Las redes neuronales de aprendizaje profundo, han mostrado su potencial en la determinación de emociones, textos e imágenes. Es por ello que existe el interés en indagar sobre el área de aprendizaje profundo en la clasificación de sílabos y compararlos con otras técnicas de evaluación para determinar si su potencial en esta área puede superarlas.

2.2 Problemática:

La clasificación de sílabos hacia áreas y subáreas del conocimiento es un proceso que se puede realizar de manera manual o automatizada. Esta tarea constituye un trabajo tedioso y un conocimiento previo de los clasificadores si se realiza de forma manual, además su evaluación esta sujeta en cierta medida al criterio del evaluador, es por ello se busca automatizar el proceso.

Al respecto de la temática, no se ha encontrado en la indagación inicial proyectos o trabajos sobre la implementación del aprendizaje profundo en esta tarea académica, ni un referente que muestre una comparación de métodos para la clasificación de texto en este ámbito educativo. Por ello, este trabajo proporcionará el uso de dicha técnica y su comparación.

2.3 Pregunta de investigación:

¿Cómo se genera una red Aprendizaje Profundo para la clasificación de planes de estudios académicos ?

¿Cuál es la arquitectura de aprendizaje profundo recomendada para la clasificación de sílabos?

2.4 Resumen:

Este trabajo será de gran relevancia para el personal académico encargado de la tarea de clasificación de sílabos hacia las áreas o subáreas del conocimiento determinadas por la UNESCO. Además de generar conocimiento sobre redes de aprendizaje profundo, se lo aplica en esta tarea académica y se contrasta con la aplicación de otras técnicas de aprendizaje automatizado en este contexto particular.

2.5 Estado de arte y marco teórico

Una tarea que aumenta la carga de trabajo del personal académico en las universidades es la comprobación de la similitud de temáticas entre sílabos, para la homologación de créditos. Este proceso se lo realiza muchas veces de forma manual y su resultado está sujeto en cierto grado al criterio del evaluador. Para cumplir esta tarea, en primera instancia se debe clasificar cada sílabo a un área y subárea del conocimiento determinado por la UNESCO, para luego seguir con el proceso de evaluación de contenido.

Para la clasificación de un sílabo dentro de un área y sub área de conocimiento, se requiere una evaluación basada en texto. Para ello, existen múltiples técnicas de aprendizaje automático, como: técnicas estadísticas, reglas, métodos lógicos simbólicos, perceptrones e instancias. (Kotsiantis, Zaharakis, & Pintelas, 2007). También, se han usado herramientas para el análisis semántico, buscando la frecuencia de términos dentro de una colección de documentos (Vuotto, Bogetti, & Fernández, 2015) y explorando la similitud de conceptos temáticos en programas académicos en la asignatura de educación física en los estados Australianos (Hyndman & Pill, 2017).

Una técnica que se ha aplicado en diferentes contextos en el tratamiento de imágenes, voz y texto, (Deng, Yu, & others, 2014) es el aprendizaje profundo, el cual permite realizar modelos computacionales compuestos por múltiples capas de procesamiento. (LeCun, Bengio, & Hinton, 2015).

Centrándonos en el campo de procesamiento de texto, el aprendizaje profundo, ha sido utilizado para presentar arquitecturas de red que permitan validar textos cortos en la selección de oraciones de respuesta, para un conjunto de preguntas (Severyn & Moschitti, 2015), así como en la comparación de semántica en un grupo de oraciones (Afzal, Wang, & Liu, 2016).

Todo este panorama, muestra la factibilidad de utilizar el aprendizaje profundo para la clasificación de los sílabos en áreas y subáreas de conocimiento designadas por la UNESCO, debido a que no existe una aplicación directa de la técnica en cuestión en esta tarea educativa, ni

antecedentes que evalúen técnicas de aprendizaje automático en contraste con el aprendizaje profundo en este contexto.

2.6 Objetivo general: Clasificar los planes de estudio académicos en base a redes neuronales de aprendizaje profundo.

2.7 Objetivos específicos:

Identificar los métodos comunes de clasificación de texto.

Identificar y seleccionar las métricas de resultados para la evaluación de algoritmos de clasificación.

- Construir un modelo computacional de aprendizaje profundo que clasifique los sílabos académicos en un área de conocimiento dictaminado por la UNESCO.
- Comparar los resultados de la red de aprendizaje profundo con los métodos previamente seleccionados.

2.8 Metodología:

- El trabajo de clasificación se realizará con aprendizaje automático supervisado, debido a que se parte desde un conjunto de datos previamente etiquetados y las áreas y subáreas donde se clasifican los sílabos ya están determinadas.

Es por ello que se partirá de una revisión bibliográfica en fuentes académicas, que permita conocer las clases de técnicas de aprendizaje automático supervisado y el aprendizaje profundo. Permitiendo evaluar algunos algoritmos de clasificación con sus métricas.

Construir el modelo de computacional de aprendizaje profundo para clasificar los sílabos académicos en un área de conocimiento dictaminado por la UNESCO.

- Documentar los resultados de la red de Aprendizaje profundo y su comparación con otras técnicas de clasificación automática supervisado.

2.9 Alcances y resultados esperados:

- Descripción de las categorías más relevantes de técnicas de aprendizaje supervisado, selección de una técnica por categoría, evaluación y presentación de un cuadro con las técnicas y sus resultados.

Construcción del modelo computacional de aprendizaje profundo y explicación de su arquitectura y resultados.

2.10 Supuestos y riesgos:

Riesgos	Probabilidad	Alternativas de solución
Que se requiera un conjunto de entrenamiento mas grande.	Media	Clasificar los sílabos de otra universidad agrandando el conjunto de entrenamiento y pruebas.
Tiempo insuficiente para el entrenamiento de la red de aprendizaje profundo.	Media	Utilizar equipos más potentes en su GPU o el saldo gratuito que google provee, para el alquiler de servicios.

2.11 Esquema tentativo:

Capítulo 1: Introducción

Capítulo 2: Marco Teórico

Capítulo 3: Construcción del modelo computacional de aprendizaje profundo y experimentación.

Capítulo 4: Evaluación de resultados

Conclusiones y trabajos futuros.

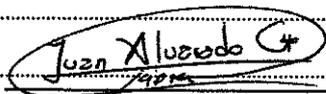
Bibliografía.

2.13 Referencias:

- Afzal, N., Wang, Y., & Liu, H. (2016). MayoNLP at SemEval-2016 Task 1: Semantic textual similarity based on lexical semantic net and deep learning semantic model. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)* (pp. 674–679).
- Deng, L., Yu, D., & others. (2014). Deep learning: methods and applications. *Foundations and Trends in Signal Processing*, 7(3--4), 197–387.
- Hyndman, B., & Pill, S. (2017). The Curriculum Analysis of Senior Education in Physical Education (CASE-PE) study. *Curriculum Perspectives*, 37(2), 147–160.
- Kotsiantis, S. B., Zaharakis, I., & Pintelas, P. (2007). Supervised machine learning: A review of classification techniques. *Emerging Artificial Intelligence Applications in Computer Engineering*, 160, 3–24.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, 521(7553), 436.
- Severyn, A., & Moschitti, A. (2015). Learning to rank short text pairs with convolutional deep neural networks. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 373–382).
- Vuotto, A., Bogetti, C., & Fernández, G. (2015). Aplicación del factor TF-IDF en el análisis semántico de una colección documental. *Biblios*, (60).

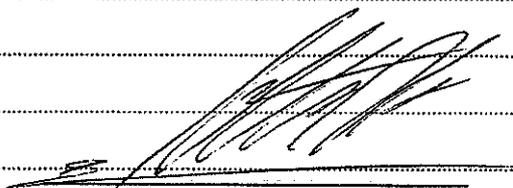
2.14 Anexos

2.15 Firma de responsabilidad (estudiante)

A handwritten signature in black ink, enclosed in a hand-drawn oval. The signature appears to read "Juan Alvarado C" with a small mark at the end. Below the signature is a horizontal line.

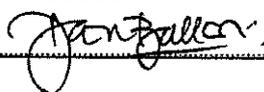
Juan Marcelo Alvarado Carrera

2.16 Firma de responsabilidad (director sugerido)

A handwritten signature in black ink, consisting of several overlapping, stylized strokes. Below the signature is a horizontal line.

Ing. Marcos Orellana Cordero

2.17 Firma de responsabilidad (Asesor Metodológico)

A handwritten signature in black ink, written in a cursive style. Below the signature is a horizontal line.

Daniela Ballari, PhD

2.18 Fecha de entrega: 09-05-2018