



FACULTAD DE INGENIERÍA
UNIVERSIDAD DE LA REPÚBLICA

INSTITUTO DE INGENIERÍA ELÉCTRICA

RECONOCIMIENTO DE PATRONES S 2017

PROYECTO FINAL

Proyecto Estándar de Fin de Curso

Luis Esteban Gonzalez Montoya

03-12-2017

Índice

1. Objetivos	III
2. Metodología	1
2.1. Selección de características	1
2.2. Clasificadores	2
2.3. Algoritmo genético-CFS	3
3. Resultados de la clasificación	4
3.1. Resultados con el set de datos completo	4
3.2. Resultados con el set de datos modificado con la selección de características	5
4. Recomendaciones	6

Índice de figuras

1. Esquema de funcionamiento de la red neuronal implementada [6]	2
2. Mapas de características con combinaciones colocadas de izquierda-derecha, arriba-abajo):	4

Índice de tablas

1. Hiperparámetros de la NN	3
2. Hiperparámetros de la CNN	3
3. Resultados de la selección de características	3
4. Porcentajes de acierto para el set de características completo	4
5. Matriz de confusión con el set de características completo	5
6. Porcentajes de acierto para diferentes sets de características obtenidos con genetic search	5
7. Porcentajes de acierto utilizando las características seleccionadas y todas las muestras	6
8. Matriz de confusión para el set reducido	6

1. Objetivos

- Utilizar algoritmos de selección de características para reducir la dimensionalidad de la base de datos "*Fashion MNIST*".
- Implementar una red neuronal y una red neuronal convolucional utilizando las librerías de python *scikit-learn* y *tensorflow*, con el fin de clasificar los patrones de la base de datos.
- Analizar los resultados obtenidos a partir de la clasificación y proponer mejoras con el fin de mejorar el porcentaje de aciertos en la clasificación.

2. Metodología

2.1. Selección de características

La selección de características en esta instancia es particularmente relevante dada la magnitud de la base de datos: 60 000 patrones de entrenamiento y 10 000 patrones de prueba, cada uno con 784 características numéricas. Con este objetivo en mente se seleccionó el artículo "*Feature extraction, feature selection and machine learning for image classification: A case study*" [5] como guía a la hora de escoger los algoritmos de selección de características a utilizar.

El artículo en cuestión utilizó una base de datos del Laboratorio de Polen de la Universidad de Newcastle, Australia con imágenes de granos de polen. La base de datos consistía de 181 muestras, 38 clases y 48 atributos numéricos por imagen.

En el artículo se analizaron nueve métodos de búsqueda en la selección de características:

- Análisis de componentes principales (PCA)
- Mejor-primero con búsqueda hacia adelante (BFF)
- Mejor-primero con búsqueda hacia atrás (BFB)
- Algoritmo evolutivo (EA)
- Algoritmo genético (GA)
- Búsqueda codiciosa (GS)
- Selección lineal hacia adelante (LFS)
- Optimización de partículas por enjambre (PSO)
- Reordenamiento por búsqueda (RRS)
- Búsqueda heurística tabú (TS)

Los subconjuntos de características obtenidos a partir de estos métodos fueron posteriormente probados utilizando doce clasificadores distintos. De acuerdo con los resultados, los autores recomiendan evitar los métodos de selección: GS, PCA y BFB, pues en ocho de los doce clasificadores estos subconjuntos produjeron los peores resultados en la clasificación; en contraposición se recomienda utilizar EA, RRS, BFF, LFS y GA, pues produjeron los mejores resultados con los distintos clasificadores.

En este caso la selección será implementada utilizando Weka [4], de los métodos de búsquedas enunciados se implementarán únicamente GA. Inicialmente se consideraron también LFS y BFF, pero con estos métodos de búsqueda los subconjuntos de características eran muy pequeños (de entre 30-50 características) aún variando los parámetros de ajuste, lo cual producía porcentajes de acierto relativamente bajos (60 %-70 %), descartándolos por completo. GA tiene la ventaja de ser el segundo método de búsqueda con mejores resultados al utilizar una red neuronal como clasificador, según el estudio ya mencionado [5]. Para la evaluación de los atributos se utilizarán dos métodos distintos, uno de filtrado y otro de envoltura con validación por clasificador.

El método de filtrado vendría siendo CFS (Correlation-based Feature Selection), pues este fue utilizado en el artículo ya mencionado con la idea de evaluar la capacidad predictiva de cada característica así como su grado de redundancia [5]. En cuanto al método de envoltura, se utilizará

el algoritmo FSE (Feature subset evaluation) con el clasificador *Naive Bayes* como clasificador de validación, dado que se busca que el tiempo de computación no sea tan alto y porque es muy comúnmente utilizado en la selección de características [3]. Cabe mencionar que el conjunto de selección corresponde al 10 % del total de muestras (6000 muestras) con el fin de reducir el costo computacional.

2.2. Clasificadores

Se decidió utilizar una red neuronal (NN) como método de clasificación principal, pues ha demostrado tener porcentajes de acierto de más del 99 % con la base de datos MNIST [1], la cual guarda una gran similitud con la base de datos utilizada.

A pesar de los buenos resultados obtenidos con las redes neuronales, para tareas más complejas el plantear redes más profundas (mayor número de capas escondidas) puede provocar sobreajuste, así como tiempos de convergencia muy altos. Las redes neuronales convolucionales (CNN) son una buena alternativa a estas y en el pasado han demostrado rendimientos récord con la base de datos MNIST [1].

Las redes neuronales convolucionales se basan en los principios de mapeo y agrupamiento para mejorar su rendimiento. El mapeo consta de un filtrado multicapa que busca identificar patrones a lo largo del set de características, esto se consigue por medio de vectores de peso de tamaño fijo (por filtro) que recorren el conjunto de características abarcando pequeños subgrupos, que posteriormente han de pasar por una unidad de rectificado lineal.

Por otro lado, el principio de agrupamiento busca disminuir el efecto que pequeñas variaciones de tamaño y orientación puedan tener sobre la clasificación. Básicamente se lleva a cabo una selección bajo un criterio estadístico (max pooling en este caso) dentro de un pequeño grupo de características.

Estas capas pueden concatenarse entre sí, volviendo la red más profunda con el fin de mejorar la clasificación. En la salida de estas capas se encuentra la capa totalmente conectada, la cual hace la función de una interfaz entre las características preprocesadas y una red neuronal clásica.

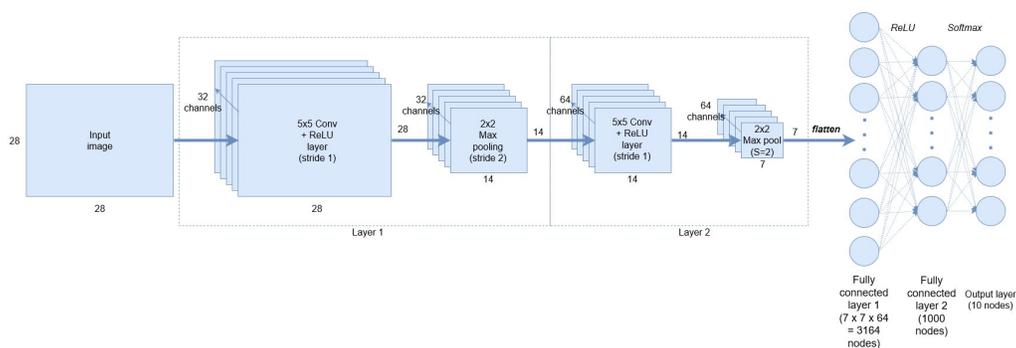


Figura 1: Esquema de funcionamiento de la red neuronal implementada [6]

En este caso particular, la red neuronal clásica será implementada por medio de la librería de python scikit-learn, utilizando específicamente el objeto *Multilayer Perceptron Classifier* (MLP). Por otro lado, la red neuronal convolucional deberá plantearse a partir de la librería tensorflow, la cuál es de una complejidad considerablemente mayor a scikit-learn. El código implementado se basa en la guía *Convolutional Neural Networks Tutorial in TensorFlow*.

Los hiperparámetros que se utilizaron en las red son una combinación de los parámetros por defecto y un conjunto de valores obtenidos a partir de varias iteraciones (razón de aprendizaje y

número de iteraciones) con el 10% del set de entrenamiento y el objeto MLP. Para la red neuronal convencional se utilizó:

Tabla 1: Hiperparámetros de la NN

Razón de aprendizaje	0.0005
Número de iteraciones	200
Tamaño del subgrupo	32
Números de unidades	128

En cuanto a la red neuronal convolucional, se utilizaron los hiperparámetros recomendados [6], variando la razón de aprendizaje. Es de resaltar que se tomó esta decisión debido a que el tiempo de computación podía volverse muy grande dependiendo de las elecciones, sin que esto se tradujese en una mejora sustancial en el número de aciertos.

Tabla 2: Hiperparámetros de la CNN

Razón de aprendizaje	0.0005
Número de iteraciones	10
Tamaño del subgrupo	32
Números de unidades	1000
Tamaño de filtro	5x5

2.3. Algoritmo genético-CFS

La probabilidad de cruce, el número máximo de generaciones y el tamaño de la población son tres de los atributos que definen el comportamiento del método de búsqueda GA [2]. Se plantearon distintas combinaciones con el fin de obtener distintos subgrupos de características. La probabilidad de mutación es otro atributo importante de este algoritmo, pero dada su naturaleza se prefirió no modificar su valor en las distintas iteraciones. Los resultados de la selección de características fueron:

Tabla 3: Resultados de la selección de características

Combinación	Tamaño de la Población	Probabilidad de cruce	Numero de generaciones	Numero de características
1	20	0.6	20	388
2	200	0.6	500	316
3	500	0.3	500	335
4	500	0.8	500	328
5	200	0.3	500	322
6	200	0.8	500	319
7	100	0.6	500	294
8	100	0.6	1000	307
9	50	0.6	1000	330
10	200	0.6	1000	303

Coloreando de negro los pixeles (o características) a utilizar en una grilla de 28x28 se obtienen los siguientes mapas de características:



Figura 2: Mapas de características con combinaciones colocadas de izquierda-derecha, arriba-abajo):

Es de resaltar cómo en la mayoría de las iteraciones el algoritmo de selección de características prioriza los píxeles en la periferia de la grilla y no en su interior, lo cual podría afirmarse que tiene sentido, pues en la mayoría de las prendas esta región tiende a ser de colores semiuniformes.

3. Resultados de la clasificación

3.1. Resultados con el set de datos completo

Ajustando el clasificador con los datos de entrenamiento completos (60 000 muestras con 784 características) se obtuvieron los siguientes resultados en la clasificación del conjunto de prueba completo:

Tabla 4: Porcentajes de acierto para el set de características completo

Clasificador	% de aciertos
Red neuronal	81.64
Red neuronal convolucional	91.31

En cuanto a la matriz de confusión, para la red neuronal se obtuvo:

Tabla 5: Matriz de confusión con el set de características completo

	T-S	PA	PO	VE	SAC	SAN	CA	ZA	BOL	BOT
T-S	803	10	18	87	7	1	44	0	29	0
PA	1	941	17	29	3	1	8	0	0	0
PO	22	1	679	11	140	0	136	0	11	0
VE	31	16	16	888	34	0	8	0	6	1
SAC	1	0	124	58	675	0	139	0	3	0
SAN	2	0	0	1	0	887	1	50	7	52
CA	221	7	114	57	88	0	492	0	20	1
ZA	0	0	0	0	0	23	0	888	1	88
BOL	1	0	9	2	4	5	17	6	956	0
BOT	0	0	0	0	0	16	0	29	1	954

T-S: T-shirt, PA: Pantalones, PO: Pullover, VE: Vestido, SAC: Saco
 SAN: Sandalias, CA: Camisa, ZA: Zapatillas, BOL: Bolso, BOT: Botas

Para el clasificador con la red neuronal convolucional no fue posible obtener la matriz de confusión debido a dificultades con la librería tensorflow.

3.2. Resultados con el set de datos modificado con la selección de características

Para los conjuntos de bits obtenidos con la selección de características descrita en la tabla (3) los porcentajes de aciertos en la clasificación del set de prueba tanto con la red neuronal, al entrenar los clasificadores con sólo el 10% del total de muestras tanto para la red neuronal, como para la red neuronal convolucional fueron los siguientes:

Tabla 6: Porcentajes de acierto para diferentes sets de características obtenidos con genetic search

Combinación	Porcentaje de aciertos	
	Red neuronal	Red neuronal convolucional
1	83.33	84.38
2	82.78	84.56
3	82.85	82.81
4	83.39	83.85
5	82.73	84.34
6	82.97	84.38
7	83.32	84.49
8	82.32	84.26
9	83.10	82.98
10	82.66	83.12

Los resultados en todos los casos son muy similares, pero dado que la combinación 1 cuenta con el tercer porcentaje de aciertos más alto para ambos clasificadores, se seleccionó este conjunto para ser re-entrenado con el set de muestras completo.

Tabla 7: Porcentajes de acierto utilizando las características seleccionadas y todas las muestras

Clasificador	% de aciertos
Red neuronal	82.34
Red neuronal convolucional	88.09

Su correspondiente matriz de confusión vendría siendo:

Tabla 8: Matriz de confusión para el set reducido

	T-S	PA	PO	VE	SAC	SAN	CA	ZA	BOL	BOT
T-S	795	11	28	102	4	1	32	1	24	1
PA	0	956	19	19	2	1	3	0	0	0
PO	18	1	744	15	181	0	31	0	10	0
VE	26	23	13	890	34	0	8	0	5	1
SAC	0	0	100	61	801	0	36	0	2	0
SAN	2	0	0	1	0	890	0	62	9	36
CA	193	6	126	86	190	0	385	0	13	1
ZA	0	0	0	0	0	23	0	899	1	77
BOL	1	0	10	2	9	1	19	6	951	1
BOT	0	0	0	0	0	9	0	67	2	922

T-S: T-shirt, PA: Pantalones, PO: Pullover, VE: Vestido, SAC: Saco
 SAN: Sandalias, CA: Camisa, ZA: Zapatillas, BOL: Bolso, BOT: Botas

4. Recomendaciones

- Implementar un algoritmo de optimización para definir los hiperparámetros de las redes neuronales.
- Mejorar el código de tensorflow con el fin de extraer la matriz de confusión de la CNN.
- Aplicar selección de características al conjunto reducido que abarque las clases pullover, saco y camisa pues sufren de los mayores índices de confusión.

Referencias

- [1] Dan Claudiu Ciresan, Ueli Meier, Luca Maria Gambardella, and Jürgen Schmidhuber. Deep big simple neural nets excel on handwritten digit recognition, 2010. *Cited on*, 80.
- [2] David E Goldberg et al. Genetic algorithms in search optimization and machine learning, 1989.
- [3] Martin Gutlein, Eibe Frank, Mark Hall, and Andreas Karwath. Large-scale attribute selection using wrappers. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 332–339. IEEE, 2009.
- [4] Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.
- [5] Madalina Cosmina Popescu and Lucian Mircea Sasu. Feature extraction, feature selection and machine learning for image classification: A case study. In *Optimization of Electrical and Electronic Equipment (OPTIM), 2014 International Conference on*, pages 968–973. IEEE, 2014.
- [6] Andy Thomas. Convolutional neural networks tutorial in tensorflow. Recuperado de: <http://adventuresinmachinelearning.com/convolutional-neural-networks-tutorial-tensorflow> 2017.