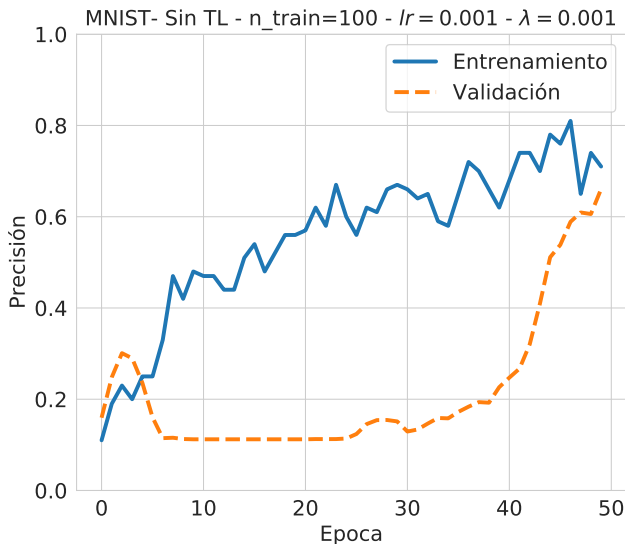


## TP5

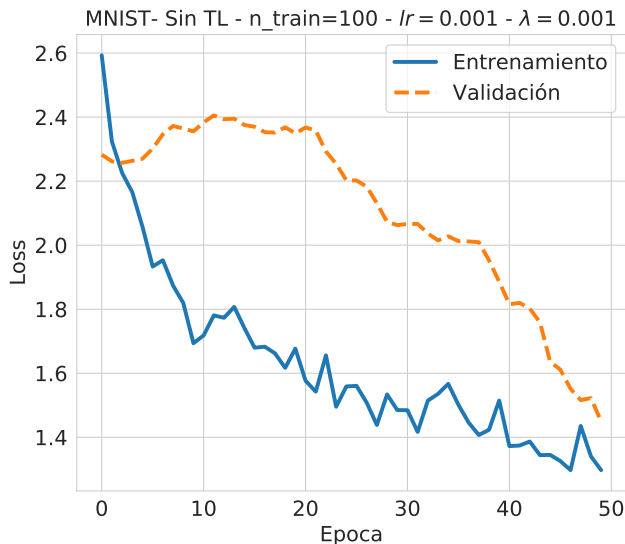
Tomás Hüttenbräucker  
Aprendizaje Profundo y Redes neuronales artificiales  
(2 de noviembre de 2020)

### EJ3 - TRANSFERENCIA DE APRENDIZAJE

#### Clasificación MNIST con pocos datos



(a) Precisión.



(b) Función de costo.

Figura 1: Predicción y función de costo para la clasificación de imágenes de MNIST del modelo TomiNet sin entrenar, con  $n_{\text{train}}=100$  datos de entrenamiento.

Se entrenó una red para clasificar la base de datos de MNIST. Para entrenamiento se usaron solamente  $n_{\text{train}}=100$  ejemplos, con el objetivo de observar el desempeño de la red con pocos datos disponibles. Para validación y testing se usaron 10.000 ejemplos. El nombre de la red es TomiNet y su arquitectura es:

Model: "TomiNet"

Layer (type)	Output Shape
conv_1 (Conv2D)	(None, 28, 28, 32)
max_pool_1 (MaxPooling2D)	(None, 14, 14, 32)
batch_norm_1 (BatchNormaliza	(None, 14, 14, 32)
conv_2 (Conv2D)	(None, 14, 14, 16)
max_pool_2 (MaxPooling2D)	(None, 7, 7, 16)
batch_norm_2 (BatchNormaliza	(None, 7, 7, 16)
flatten_6 (Flatten)	(None, 784)
dense (Dense)	(None, 9)

Total params: 12,201

Trainable params: 12,105

Non-trainable params: 96

Como se observa en la figura 1, aunque se usaron solo 100 datos de entrenamiento, el modelo consigue obtener una precisión considerable en la clasificación de los 10.000 datos de validación. Esto es así porque el problema de clasificación de MNIST es un problema relativamente sencillo. Cabe destacar que, al usar solo 100 datos de entrenamiento, el modelo es muy sensible a estos, en varios entrenamientos con diferentes sets de datos de entrenamiento se obtuvieron apreciablemente diferentes.

#### Fashion MNIST a MNIST.

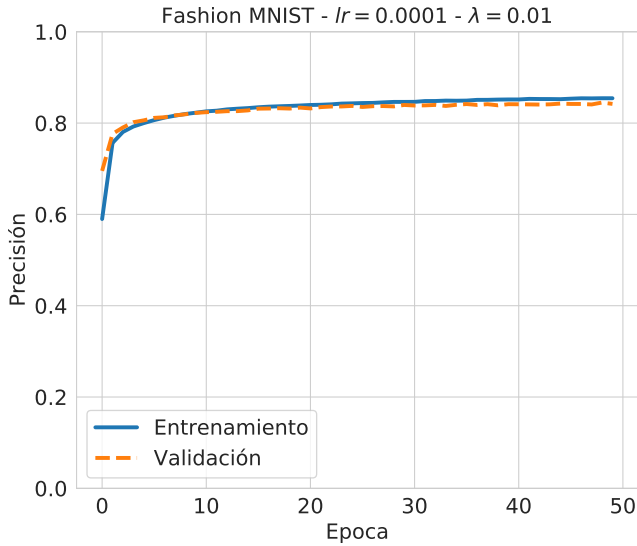
Se entrenó una red con la arquitectura de TomiNet para clasificar el dataset Fashion MNIST, que esta compuesto por 60.000 imágenes de  $28 \times 28$  píxeles blanco y negro, usando 40.000 para entrenar, 10.000 como validación y 10.000 de testing.

En la figura 2 se muestra la evolución del modelo. Aunque con otro modelo se podría conseguir un mejor resultado, lo relevante de este ejercicio es observar si se puede transferir lo aprendido a una nueva red.

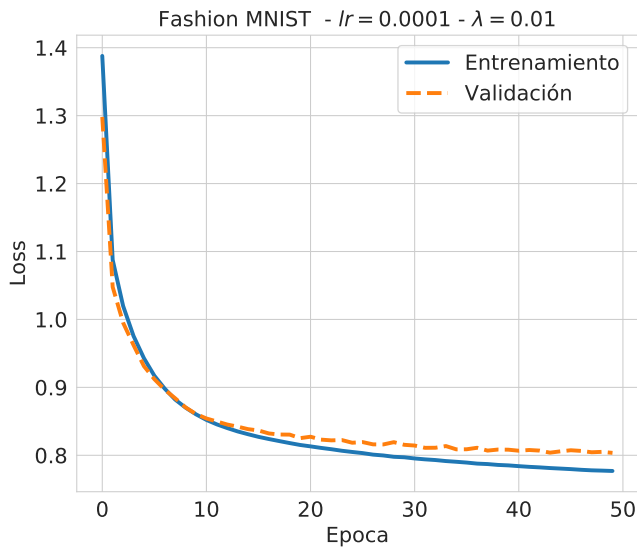
A continuación, se aplicó transferencia de aprendizaje de la red entrenada con los datos de Fashion MNIST para la clasificación de datos MNIST. Para esto se usó el modelo ya entrenado con los datos de Fashion MNIST, cambiando la ultima capa para adecuarla a la nueva clasificación. Se utilizó un esquema en el que por 35 épocas se entrenó solo a la capa densa de salida del modelo, sin tocar los pesos previos y luego se realizó *fine tuning* por 15 épocas, con un learning rate 10 veces menor.

En la figura 3 se muestra el aprendizaje del modelo al

usar Transfer Learning. Aunque en el principio del entrenamiento parece ser mucho mejor que el mismo modelo sin transferencia de aprendizaje (observando los datos de validación), termina teniendo un rendimiento similar. Esto puede deberse a que las imágenes de Fashion MNIST no tienen las mismas características que las imágenes de MNIST y por eso no representa una mejora relevante el uso de transferencia de aprendizaje. A pesar de esto, el modelo si presenta una mayor consistencia respecto a los datos de entrenamiento, mostrándose mas independiente a los posibles cambios de estos.

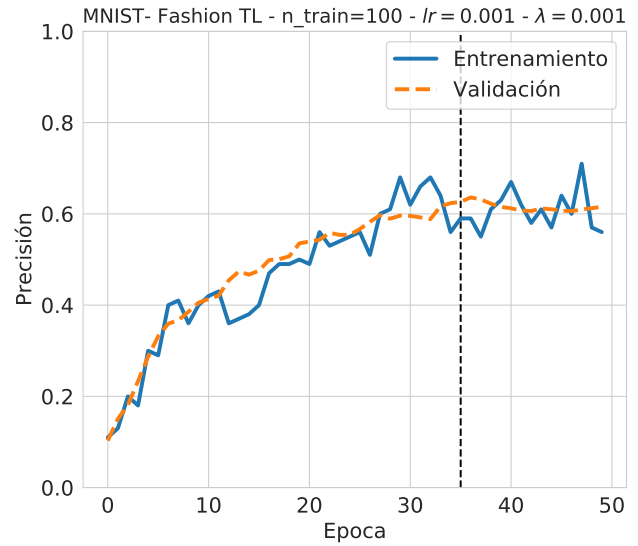


(a) Precisión.

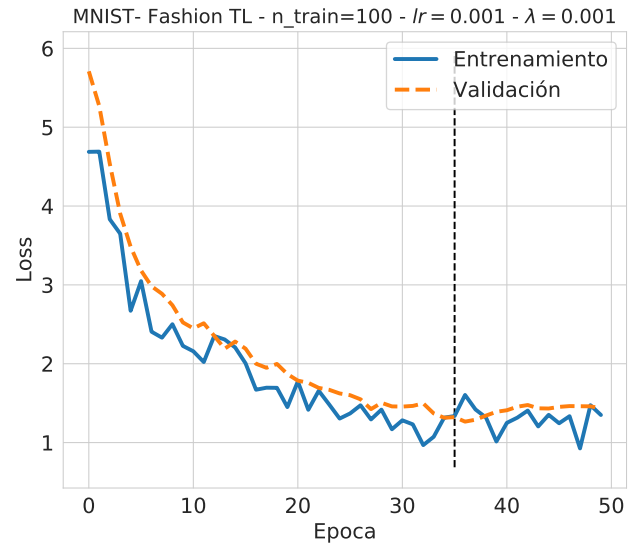


(b) Función de costo.

Figura 2: Predicción y función de costo del modelo utilizado para la clasificación de imágenes de Fashion MNIST.



(a) Precisión.



(b) Función de costo.

Figura 3: Predicción y función de costo para la clasificación de imágenes de MNIST del modelo TomiNet previamente entrenado sobre Fashion MNIST, aplicando transferencia de aprendizaje con  $n_{train}=100$  datos de entrenamiento.

### MNIST Letters a MNIST

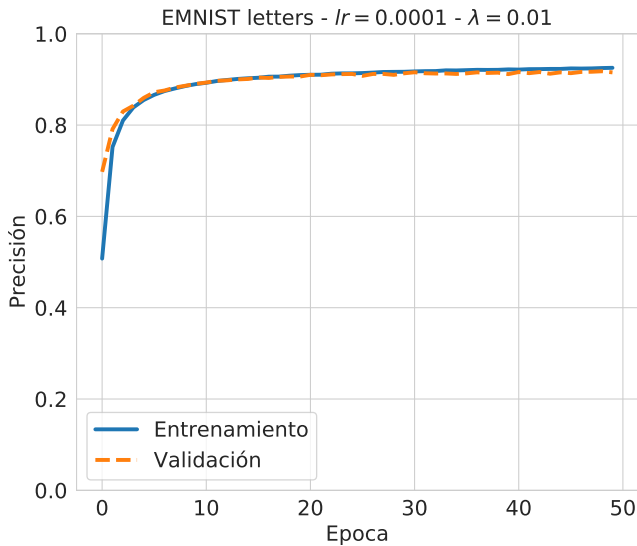
Se entrenó nuevamente el modelo TomiNet pero esta vez para clasificar los datos de MNIST letters, que incluye 60.000 imágenes de  $28 \times 28$  en blanco y negro de letras del alfabeto.

En la figura 4 se muestra el aprendizaje del modelo. Nuevamente, el objetivo no es analizarlo, si no simplemente observar que clasifica las imágenes de forma aceptable.

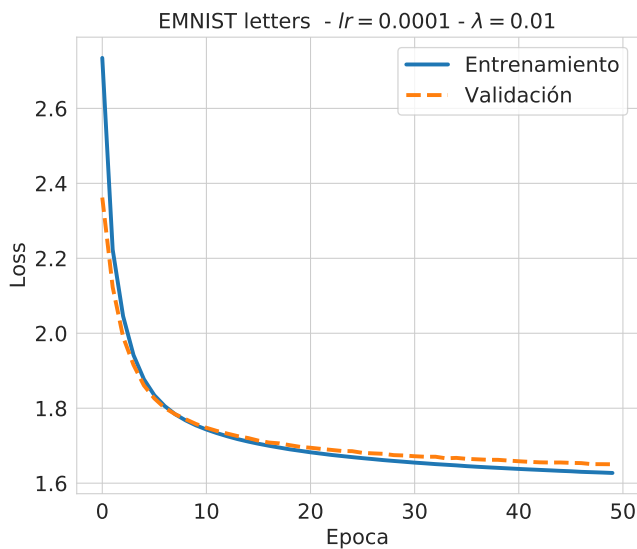
Una vez entrenado el modelo, se procedió a realizar la transferencia de aprendizaje para clasificar las imágenes de MNIST. Para esto se copio el modelo y se cargo con los mismos pesos, cambiando la ultima capa para la nueva clasificación. Se entrenó por 35 épocas solo la capa de

salida y por 15 épocas con un learning rate 10 veces mas pequeño a todo el modelo. Solo se usaron 100 datos de entrenamiento.

En la figura 5 se muestra la evolución del modelo. Se ve que aunque haya solo  $n_{train}=100$  datos de entrenamiento, el modelo consigue clasificar bastante bien el set de datos y, en particular, bastante mejor que la misma red sin transferencia de aprendizaje o con transferencia de aprendizaje de Fashion MNIST. Esto puede deberse a que las letras y los números tienen mucho más en común que la ropa y los números, entonces la transferencia de aprendizaje es mucho más efectiva.

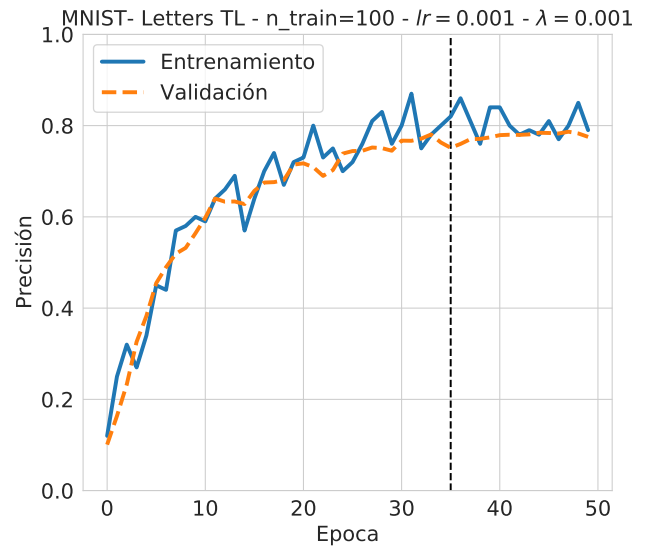


(a) Precisión.

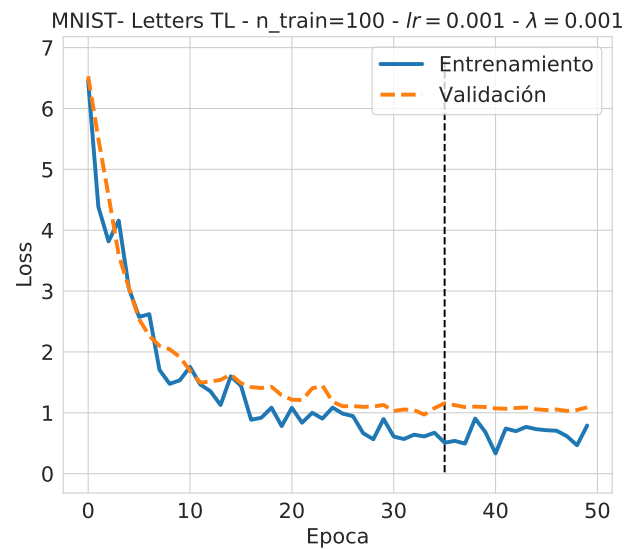


(b) Función de costo.

Figura 4: Predicción y función de costo para la clasificación de imágenes de MNIST letters del modelo TomiNet.



(a) Precisión.



(b) Función de costo.

Figura 5: Predicción y función de costo para la clasificación de imágenes de MNIST del modelo TomiNet previamente entrenado sobre MNIST letters, aplicando transferencia de aprendizaje con  $n_{train}=100$  datos de entrenamiento.

### Comparación

Con los resultados obtenidos sobre los datos de testing, mostrados en 1 y con los resultados obtenidos anteriormente, se puede concluir que al aplicar transferencia de aprendizaje, lo mejor es usar imágenes que tengas características en común con las de entrenamiento previo de la red.

- Al usar la red sin entrenamiento previo sobre los 100 datos de entrenamiento se obtuvieron resultados mejores de los esperados, pero que variaban mucho de acuerdo a la ‘bondad’ del set de entrenamiento, obteniendo resultados variables de acuerdo a este.

Modelo	Acc. Test
Sin TL	0.67
TL- Fashion	0.63
TL-Letters	0.77

Tabla 1: Precisión sobre los datos de testing de los diferentes modelos.

- Al usar Fashion MNIST como entrenamiento previo, no se obtuvieron resultados mucho mejores que al no usar entrenamiento previo, sin embargo, se consiguieron resultados más consistentes y más independientes del set de entrenamiento.
- Al usar MNIST letters como entrenamiento previo para luego clasificar MNIST, se obtuvieron los mejores resultados, lo que tiene sentido ya que las letras y los números tienen mucho en común.

Cabe destacar que debido a que solo se usan 100 datos de entrada para clasificar 10 categorías, estos toman una relevancia importante. Sería mejor realizar estadística sobre los resultados, entrenando varias veces los modelos y observando como se comportan ante variaciones en los datos de entrada, o quizás elegir un conjunto representativo del total y mantenerlo para todos los modelos.

#### EJ4 - VISUALIZACIÓN DE LA RED

Se entreno un modelo para clasificar las imágenes del dataset MNIST con el objetivo de poder visualizarlo. La arquitectura del modelo usado fue:

Model: "TomiNet"

Layer (type)	Output Shape
conv_1 (Conv2D)	(None, 28, 28, 32)
max_pool_1 (MaxPooling2D)	(None, 14, 14, 32)
batch_norm_1 (BatchNormaliza	(None, 14, 14, 32)
conv_2 (Conv2D)	(None, 14, 14, 16)
max_pool_2 (MaxPooling2D)	(None, 7, 7, 16)
batch_norm_2 (BatchNormaliza	(None, 7, 7, 16)
flatten_4 (Flatten)	(None, 784)
dense (Dense)	(None, 10)

Para visualizar el modelo, se entrenaron imágenes de entrada para que maximicen el valor medio de la activación de cada uno de los filtros de las diferentes capas.

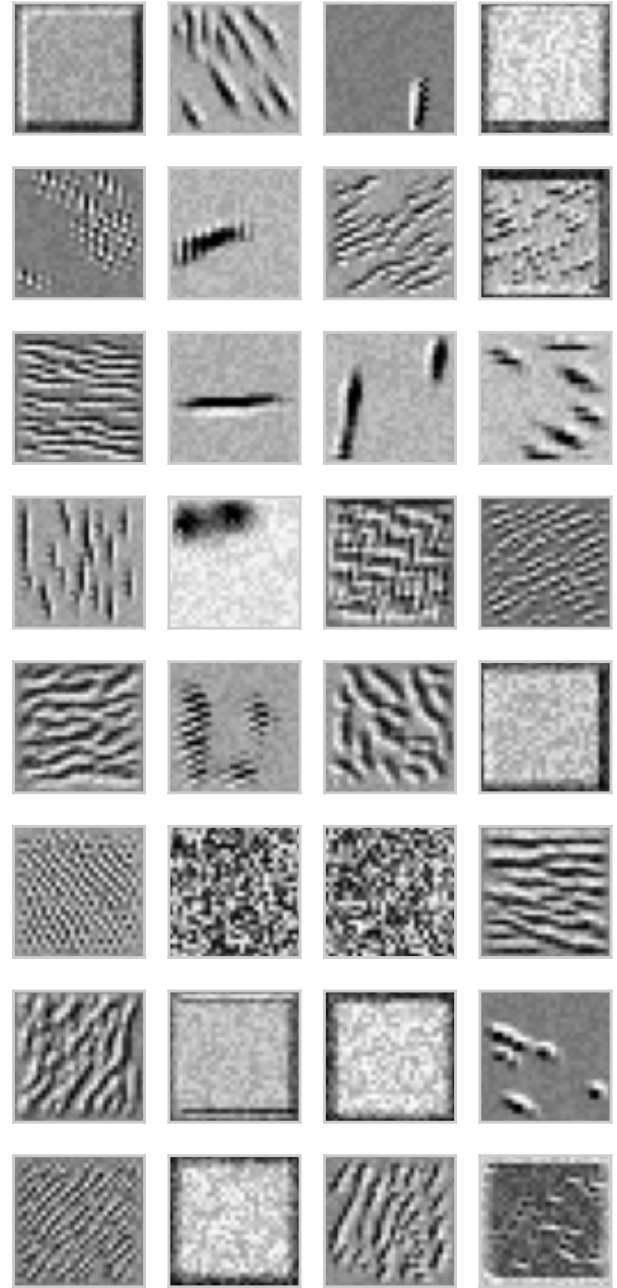


Figura 6: Entradas que maximizan las activaciones de los filtros convolucionales de la primera capa.

En las figuras 6 y 7 se muestran las entradas que maximizan las activaciones de cada uno de los filtros de las dos capas convolucionales del modelo.

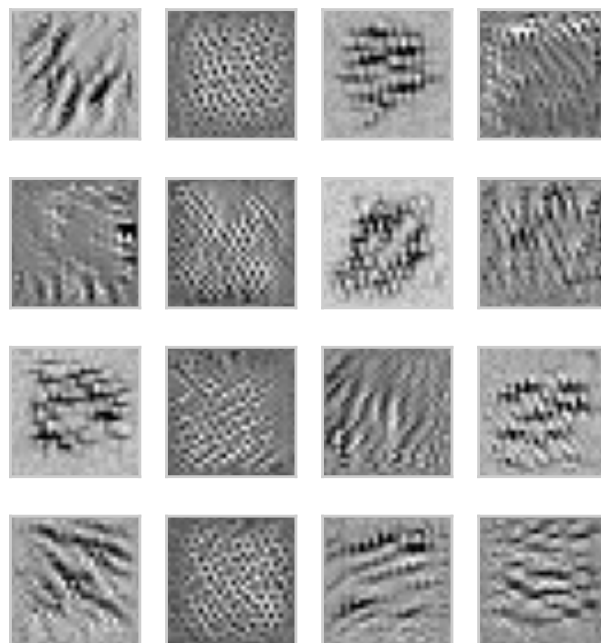


Figura 7: Entradas que maximizan las activaciones de los filtros convolucionales de la segunda capa.