

Redes Neuronales Convolucionales

Práctica 4

Ejercicio 1. Convoluciones 1D En el archivo “Práctica 4 - Convoluciones 1D.ipynb” encontrarás un notebook donde deberás implementar algunas variantes de las convoluciones en 1D, es decir, sobre un vector.

Ejercicio 2. Convoluciones 2D En el archivo “Práctica 4 - Convoluciones 2D.ipynb” encontrarás un notebook donde deberás implementar algunas variantes de las convoluciones en 2D, es decir, sobre una matriz o imagen.

Ejercicio 3. CNN en MNIST Diseñe un modelo de Redes Neuronales Convolucionales (CNN) para clasificar el conjunto de datos MNIST visto en la práctica 3. Pruebe 5 modelos distintos, variando los siguientes hiperparámetros:

- Cantidad de capas Conv
- Cantidad de feature maps
- Stride, padding y kernel_size de las convoluciones
- Funciones de activación
- Uso de capas Max Pooling.

Ejercicio 4. Optimización de Hiperparámetros en MNIST. En este ejercicio vamos a probar variando los hiperparámetros de la red de forma sistemática para encontrar una arquitectura que se ajuste mejor al problema, es decir, la arquitectura más simple que obtenga un mejor accuracy en test. Comience utilizando una red de tres capas con la siguiente arquitectura:

Capa Convolutiva (Conv2D) - Capa Max Pooling (MaxPooling2D) - Capa de (Regresión Logística (Dense))

Varíe sistemáticamente los parámetros de la tabla y tome nota del accuracy obtenido en train y test y la cantidad de épocas hasta que converge el modelo. Podemos considerar que el modelo converge si el accuracy de test alcanza su máximo y deja de cambiar por 5 épocas. Para eso puede utilizar la clase **EarlyStopping** de Keras:

```
es = EarlyStopping(monitor='val_acc', mode='min', patience=5, verbose=1)
```

E incluya el objeto como callback en el método fit:

```
history = model.fit(trainX, trainy, validation_split=0.1, epochs=4000, verbose=0, callbacks=[es])
```

El parámetro *patience* le indica a **EarlyStopping** cuantas iteraciones sin cambio esperar.

Para realizar los experimentos, NO modifique a mano los hiperparámetros cada vez que quiera probar una nueva combinación. En lugar de eso, escriba un script principal que:

1. Genere las combinaciones de hiperparámetros a probar
2. Por cada combinación, llame a otro script o función para entrenar y probar el modelo

3. Anote los resultados de cada modelo (junto con los hiperparámetros utilizados) en una tabla. Recomendamos guardar los resultados en un archivo .csv para luego inspeccionarlos manualmente.

Feature Maps en capa Conv2D	Tam. del kernel (kernel_size)	Stride en capa Conv2D	Max Pooling (MP)	Stride en MP	Función de activación	Épocas	Acc. en Train	Acc. en Test
4	3x3	1	Si	1	ReLU			
16	3x3	1	Si	1	ReLU			
64	3x3	1	Si	1	ReLU			
128	3x3	1	Si	1	ReLU			
4	7x7	1	Si	1	ReLU			
16	7x7	1	Si	1	ReLU			
64	7x7	1	Si	1	ReLU			
128	7x7	1	Si	1	ReLU			
64	3x3	2	No	-	ReLU			
64	3x3	3	No	-	ReLU			
64	3x3	1	Si	-	TanH			
64	3x3	1	Si	-	Sigmoide			

Ejercicio 5. CNN en CIFAR10 Repita el ejercicio 3 para el conjunto de datos CIFAR10, también visto en la práctica anterior. El entrenamiento para CIFAR10 generalmente tarda más tiempo que para MNIST debido a que las imágenes tienen aproximadamente tres veces la dimensionalidad y además mayor complejidad. Por ende, pruebe sólo algunas arquitecturas distintas.

Ejercicio 6. Competencia de clasificación de imágenes de radiografías de tórax. Hemos creado una competencia de Kaggle exclusiva para los alumnos de la materia.

El conjunto de datos contiene imágenes de pacientes con radiografías normales y con neumonía. Tu objetivo es entrenar un modelo con el F-Score más alto en el conjunto de test. Para evitar trampas, las etiquetas del conjunto de test estarán ocultas hasta el final de la competencia. Mientras, tendrás que utilizar un conjunto de *validación* (que sí está etiquetado) para determinar si tu modelo es bueno.

Kaggle te permite utilizar *kernels* similares a los notebooks de Jupyter, con hasta 30 horas de GPU por mes/usuario para entrenar modelos. También podés entrenar en tu propia computadora u otra plataforma como Google Collab y subir los resultados.

El siguiente enlace privado te permite anotarte en la competencia (sólo para alumnos de la materia):

<https://www.kaggle.com/t/85b61df13c79401a92df9272b08e9c46>

La competencia finalizará el lunes 4 de noviembre. Ese día discutiremos los resultados de la misma en la clase de repaso.

La competencia tiene un premio simbólico, será algo comestible y de chocolate. El objetivo de la misma es incentivar el trabajo en equipo y obtener el conocimiento práctico necesario para enfrentarse a un problema de aprendizaje automático.

Recomendamos encarecidamente que formen equipos para participar en la competencia. Pueden formar equipos de tamaño ilimitado, pero cada equipo sólo puede presentar dos conjuntos de predicciones finales.

El resto de los detalles, así como un notebook con un modelo de ejemplo, se encuentra en la página de la competencia.