

REDES NEURONALES 2024

Trabajo Final Integrador

Autoencoder y clasificador convolucional sobre Fashion-MNIST

Nota:

- Entreguen el trabajo final integrador **sólo** en formato **.pdf**. Si desean pueden enviar la notebook, pero por separado.
- El **.pdf** no puede tener más de cuatro (4) páginas.

PARTE 1) Red neuronal autoencoder convolucional de varias capas

Defina y cree una red neuronal autoencoder convolucional. El autoencoder debe poseer dos módulos, un encoder y un decoder. El **encoder** tienen que tener al menos dos capas convolucionales 2D y una lineal. El **decoder** tiene que realizar una transformación aproximadamente inversa, por ejemplo, utilizando primero una capa lineal y luego dos convolucionales traspuestas.

Recuerde incluir dropout, si es que lo considera necesario, y elegir adecuadamente el tipo de unidades de activación de la capa de salida. No utilice dropout en la capa de salida. Porqué?

PARTE 2) Entrenando el autoencoder

2.1) Implemente en una función **train_loop** un loop que itere en modo entrenamiento sobre los **batches** o lotes de una época de entrenamiento.

2.2) Implemente en una función **eval_loop** un loop que itere en modo evaluación sobre los batches de una época de entrenamiento.

2.3) Inicialice los **DataLoaders** sobre el conjunto de entrenamiento y el conjunto de validación, usando **batches** de 100 ejemplos. **AYUDA:** Crear nuevas clases derivadas de la clase **Dataset** que sirvan para entrenar autoencoders, i.e. en donde tanto el input como el output sean la misma imagen.

2.4) Cree una función de pérdida usando el **Error Cuadrático Medio (ECM)**.

2.5) Cree un optimizador con un learning rate igual a 10^{-3} . Pruebe con **ADAM**.

2.6) Cree una instancia del modelo autoencoder.

2.7) Especifique en que dispositivo (**device**) va a trabajar. Lo hará en una **CPU**, en una **GPU** o una **TPU**?

2.8) Implemente un loop que itere sobre épocas de entrenamiento. Este loop debe guardar en listas correspondientes, y en función de las épocas, los promedios del ECM sobre los conjuntos de entrenamiento y validación, incluyendo los valores incorrectamente estimados del ECM sobre el conjunto de entrenamiento. **IMPORTANTE:** No olvide copiar los batches al dispositivo de trabajo.

2.9) Entrene y valide el modelo.

2.10) Grafique los distintos valores del ECM en función de las épocas de entrenamiento. Cual es el número óptimo de épocas de entrenamiento? Discuta y comente.

2.11) Grafique, comparativamente, algunas de las imagenes a predecir vs las imagenes predichas por el modelo entrenado.

2.12) Repita utilizando variaciones de los hiperparámetros del autencoder. Por ejemplo, utilizando capas lineales de otros tamaños, valores distintos de dropout, otros optimizadores, distintos learning rates, distintas transformaciones convolucionales y convolucionales traspuestas, etc. Que valores de estos hiperparámetros considera los más convenientes? Porqué?

PARTE 3) Definiendo y entrenando un clasificador convolucional reutilizando el encoder

- 3.1)** Defina y cree un clasificador convolucional, agregando una **capa clasificadora** al **encoder** del autoencoder previamente entrenado.
- 3.2)** Reimplemente las funciones de entrenamiento, teniendo en cuenta que ahora debe incluir el cálculo de precisiones.
- 3.3)** Cree una función de pérdida usando la **Cross Entropy Loss (CEL)**.
- 3.4)** Cree una instancia del modelo clasificador.
- 3.5)** Entrene y valide el modelo.
- 3.6)** Grafique los distintos valores de la CEL y la precisión calculados, en función de las épocas de entrenamiento.
- 3.7)** Utilice el conjunto de validación para calcular una **Matriz de confusión**. Grafíquela y comente los resultados.

PARTE 4) Preentrenamiento

Modifique el optimizador para que sólo reentrene los parámetros de la capa clasificadora, dejando los parámetros de la capa codificadora tal como vienen entrenada del el autoencoder convolucional. Repita los experimentos de la parte 3. Que observa? Comente

Ayuda: Se recomienda guardar en archivos los pesos de las distintas capas de las redes entrenadas para que puedan ser reutilizadas.