

Tarea N°2:

Redes Neuronales

Cesar Hormazabal Galleguillos

Escuela de Ingeniería, Universidad de O'Higgins

31, octubre, 2023

Abstract—Como objetivo se tiene el utilizar distintos modelos de entrenamientos de redes neuronales para analizar su desempeño en la clasificación en imágenes . Se utilizo la base de datos "digitstrain" y "digitstest" para comparar los modelos de redes aplicados .

I. INTRODUCCIÓN

El objetivo de esta tarea es utilizar redes neuronales en un problema de clasificación de dígitos. Se utilizará el conjunto de datos Optical Recognition of Handwritten Digits Data Set. Este conjunto tiene 64 características, con 10 clases y 5620 muestras en total.

Las redes a ser entrenadas tienen la siguiente estructura: capa de entrada de dimensionalidad 64 (correspondiente a los datos de entrada), capas ocultas (una o dos) y capa de salida con 10 neuronas y función de activación .

Se usara PyTorch para entrenar y validar la red neuronal que implementa el clasificador de dígitos. Se analizará los efectos de cambiar el tamaño de la red (número de capas ocultas y de neuronas en estas capas) y la función de activación.

II. MARCO TEÓRICO

Describiendo brevemente el funcionamiento de los métodos antes de que se utilicen tenemos:

A. Red neuronal

Es un modelo computacional inspirado en la estructura y el funcionamiento del cerebro humano. Es una técnica de aprendizaje automático (machine learning) que se utilizara en el desarrollo de esta tarea, ademas sirve para abordar una variedad de problemas, como reconocimiento de patrones, clasificación, regresión, procesamiento de lenguaje natural y más.

B. Estructura de una red neuronal

La estructura basica de una red neuronal consta de 3 tipos de capas, Capa de Entrada, Capas Ocultas y Capa de Salida. Las neuronas en cada capa están interconectadas y tienen pesos que se ajustan durante el entrenamiento para mejorar la precisión de las predicciones. Cada conexión entre neuronas tiene un peso que indica la importancia de esa conexión particular en el proceso de aprendizaje.

C. Funcion activacion

En el caso de esta tarea se aplicaran ReLU y Tanh, y una función de activación es una función matemática que determina la salida de una neurona. La función de activación introduce no linealidades en el modelo, lo que permite a las redes neuronales aprender patrones complejos en los datos.

D. Conjunto de datos

En el aprendizaje supervisado, los datos se dividen típicamente en tres conjuntos: datos de entrenamiento, datos de validación y datos de prueba.

La división adecuada de los datos en estos tres conjuntos es fundamental para evitar el sobreajuste y para garantizar que el modelo generalice bien a datos no vistos.

III. METODOLOGÍA

Se implemento un código que lee las base de datos, guardando los datos de 1digitstrain y 1digitstest en dataframes para ser trabajado. Luego se aplica un split al dataframe que leyo 1digitstrain para dividir así los datos en entrenamiento y validacion. Finalmente se normalizan los datos y quedan listos para ser trabajados.

A continuacion se crean los modelos de redes neuronales definiendolos 1 a 1, de acuerdo a lo que se pide en el enunciado.

Luego de que se crean los datasets y dataloaders para pytorch. Como se pide entrenar una red neuronal para cada uno de los casos definidos anteriormente(6), se define 1 a funcion modeloDeEntrenamiento(modelX).

Describiendo el funcionamiento de la funcion, esta empieza por recibir el modelo en el que trabajaremos, y parte indicando que corran el modelo con gpu. Inmediatamente se definen la funcion perdida y optimizador que se usaran. seguido de las variables mejorperdida paciencia y sinMejoras, que seran variables que se utilizan mas adelante para el early stopping. Se implementa además una función time() que guarda los tiempos de ejecución del programa para mostrar cuando tardo.

Apartir de la función time empieza la creación de los arreglos que usamos para guardar las perdidas de cada época, su accuracy y lo que duro el entrenamiento en épocas.

Luego de esto un loop for se encarga de ejecutar las 1000 épocas que debe durar la ejecución, y dentro de cada época

se crean arreglos que guardan los datos que resultan de cada entrenamiento en cada batch tanto para el dataloader train como para el val. Siendo aplicado la predicción en el conjunto correspondiente.

También se implementa un early stopping que guardara la mejor perdida junto por encargarse de guardar este modelo, y así como su nombre lo dice detendrá la ejecución del loop encargado de las 100 épocas si se alcanza el valor de la paciencia

Luego se guarda el tiempo que tarda y la época en la que finalizo, junto con graficar el loss de entrenamiento y validación.

Finalmente se calcula la matriz confusión y el accuracy en el conjunto de entrenamiento cargando el mejor modelo que se guardo en el earlystopping, cargando además los datos correspondientes en train y val. Esto para que torch.nograd guarde en arreglos los valores que utilizaremos para la matriz de confusión, pudiendo así finalmente tener todos los datos para ser plotados.

Para finalizar basta con utilizar la función modeloDeEntrenamiento definida anteriormente en cada uno de los modos para terminar el desarrollo

IV. ANÁLISIS DE RESULTADOS

A. Efectos de variar cantidad de neuronas en capa oculta

En general hará que el modelo sea mas lento durante el entrenamiento dado que se deben realizar mas cálculos, pero es el aumento de representación de la red neuronal o que vea a una mejora en el desempeño

B. Efectos de variar cantidad de capas ocultas

Una mayor cantidad de capas ocultas aumenta su representación dado que permite que la red aprenda características y abstracciones mas complejas a medida que se profundiza, con un mayor riesgo de sobreajuste

C. Explicar el efecto de la función de activación y como esto afecta el desempeño de la red

Como aprendimos el efecto de las funciones de activación es introducir no linealidades, que permiten a la red aprender patrones complejos y representaciones mas abstractas. Lo anterior impacta mejorando la capacidad de la red para modelar relaciones no lineales, mejorando la velocidad de convergencia y evitando sobre ajuste.

D. Analizar los siguientes puntos: tiempos de entrenamiento, matrices de confusión y accuracies

Tiempo de entrenamiento:

Es importante analizar el tiempo que una red neuronal tarda en entrenarse el conjunto de datos dado que tiempos mas cortos son mas beneficiosos, pero a medida que la red se e agregan mas capas ocultas o neuronas, los tiempos se extenderán debido a que se requerirán mayor numero de cálculos y/o actualizaciones de parámetros.

Matrices de Confusión:

Son las matrices que proporcionan información sobre el rendimiento del modelo permitiendo evaluar verdaderos positivos, negativos falsos positivos y falsos negativos. Se establece que diferencias significativas entre las clases pueden indicar desequilibrios en los datos o problemas en la arquitectura de la red.

Accuracy:

Es la proporción de predicciones correctas sobre el total de predicciones realizadas por el modelo en el conjunto de datos de validación y es una medida común para evaluar el desempeño de un modelo de clasificación.

E. Analizar la matriz de confusión y el accuracy en el conjunto de prueba, respecto a los obtenidos en el conjunto de validación.

Matriz de confusión en el conjunto de prueba:

En este caso el modelo muestra como se comporta con datos nuevos (conjunto de prueba) en comparación con los datos vistos en los datos de entrenamiento. Esta debería mostrar la distribución de las clasificaciones realizadas por el modelo en el conjunto de prueba, identificando así si existen diferencias en los patrones de clasificación entre ambos conjuntos

Accuracy en el conjunto de prueba:

Comparando el accuracy en el conjunto de prueba con el de validación se proporciona una medida de como el modelo se comporta en datos no vistos indicándonos que tan bien esta clasificando el modelo las muestras nuevas. La idea es evaluar si el modelo mantiene un desempeño similar en ambos conjuntos.

V. ANÁLISIS

Es entonces necesario tener en cuenta el contexto de los datos a estudiar dado que en los distintos modelos de redes neuronales se tienen fortalezas y debilidades, por lo que se espera que se elijan de acuerdo a los datos que se estén por analizar y el objetivo que se busca en este.

Además se deberá considerar si incluir capas adicionales producirá un efecto por lo que puede ser recomendable evaluar los resultados observando las distintas matrices y matrices de confusión. Por ultimo se debe tener en cuenta los tiempos de entrenamiento junto con las matrices que comentamos para evaluar el desempeño de los modelos en términos de precisión de clasificación y capacidad de generalización.

VI. CONCLUSIONES GENERALES

Basándonos en todo lo expuesto anteriormente sobre los resultados de los modelos sobre la base de datos tenemos que las redes neuronales generan clasificaciones con mayores niveles de abstracción. Pudiendo ajustar las distintas redes neuronales de acuerdo a las necesidades (ya sea relu, tanh, agregando mas capas ocultas y/o neuronas), obteniendo una predicción que capture características mas complejas

La adecuada selección del modelo de red neuronal, junto con la la revisión detallada de los resultados son fundamentales para obtener conclusiones significativas de un conjunto de datos. Además el análisis las redes puede proporcionar información valiosa y ayudar a identificar patrones y tendencias que no son evidentes a simple vista.

VII. BIBLIOGRAFÍA

Ayudantía 3:

https://ucampus.uoh.cl/uoh/2023/2/COM4402/1/material_docente/detalle?id=2511761

ayudantia 4:

https://ucampus.uoh.cl/uoh/2023/2/COM4402/1/material_docente/detalle?id=2522245

Entrenamiento de redes neuronales - Métricas:

<https://logongas.es/doku.php?id=clase:iabd:pia:2eval:tema07.metrics>