

ELEM0012 - Inteligencia Computacional

TAREA 1

Profesor: Jorge R. Vergara

Depto. de Informática y Computación
Segundo semestre 2024

1. Parte Teórica

1. ¿Cuál de las siguientes alternativas describe de mejor manera un dataframe de pandas?
 - a) Estructura de datos bidimensionales, similar a una tabla de base de datos
 - b) Matriz de datos numérica pero eficiente
 - c) Lista de datos etiquetados
 - d) Estructura Multi-dimensional de consultas
2. ¿Qué significa un dato NaN en Python?
3. Indique ejemplos de medidas o datos discretos
4. ¿Qué son los siguientes elementos? color de ojo de una persona, temperatura, peso y altura
5. ¿Cuál de las siguientes líneas en lenguaje Python, darían un error por case sensitivo?
 - a) $a = 3$
 $b = 5 + a$
 - b) $a = 0$
 $b = 5/a$
 - c) $a = "hola"$
 $b = a + 3$
 - d) $a = [1, 2, 3]$
 $b = a \cdot 2$
6. ¿Cuál de los siguientes problemas no deberían ser resueltos con el método append?
 - a) Insertar un elemento en una posición específica de un arreglo
 - b) Agregar elementos a un arreglo sin importar la posición
 - c) Agregar elementos a una matriz sin importar la posición
 - d) Agregar listas a un campo de un diccionario
7. Encuentra la salida del siguiente programa

```
a=1
while True:
    if a%9==0:
        break
    print(a)
    a +=1
```

1

2

3

4

5

6

8. Sea una arquitectura de red como la mostrada en la Figura 1. La función de activación para todas las neuronas es sigmoide.
- Determine la salida de la red neuronal (y) en función de los pesos y de las entradas x_1 y x_2
 - Determine las reglas de actualización para cada uno de los pesos en función de y , las entradas x_1 y x_2 , la tasa de aprendizaje μ y la salida deseada d . Para este ejercicio utilice como función de costo el Error Cuadrático Medio (MSE)
 - Realice tres actualizaciones de pesos al presentar las muestras $x = [(2, -2), (1, 0, 3, 2), (-0, 3, 0, 3)]$ con salida deseada $d = [1, 0, 1]$, suponiendo un vector de pesos inicial $w = (w_{1A}, w_{2A}, w_{AB}, w_{1B}, w_{2B}) = (0, 1, 1, 1, -1)$ y una tasa de aprendizaje $\mu = 0,1$.

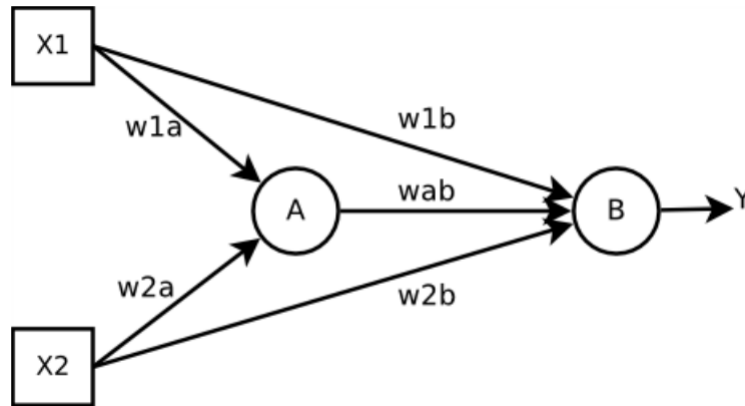


Figura 1: Esquema red neuronal.

2. Parte Experimental

Esta tarea tiene por objetivo realizar un clasificador de dibujos manuscritos usando una neuronal MLP y una red Convolutiva. Para esto, se pide que cree en pytorch y entrene sus redes usando la base de datos QUICK, DRAW¹², la cual contiene 50 millones de dibujos en 345 categorías. Cada muestra de la base de datos (Figura 2) es un mapa de 51×51 píxeles (2601 píxeles).



Figura 2: Muestras de la base de datos QUICK, DRAW.

¹<https://github.com/googlecreativelab/quickdraw-dataset>

²<https://quickdraw.withgoogle.com/>

Se pide entrenar 2 redes: una MLP de arquitectura $2601 - N - 10$, donde N es el número de neuronas de la capa oculta, y una CNN con alguna configuración de su preferencia. Usted debe seleccionar solo 5 de las 345 clases de la base de datos. Para esto, utilice el siguiente código reemplazando su RUT (con dígito verificador y sin puntos ni guión, p.e. 12.345.678-9 \rightarrow 123456789) en la variable:

```

1      import numpy as np
2      import pandas as pd
3      RUT =123456789
4      np.random.seed(RUT)
5      nameClass = pd.read_csv('https://raw.githubusercontent.com/ +
6      'googlecreativelab/quickdraw-dataset/master/categories.txt',header=None)
7      idxClass = np.random.permutation(nameClass.shape[0])[:5]
8      print('Sus Clases son:')
9      print(nameClass.values[idxClass])

```

La función de activación de la red MLP que se usará para las capas ocultas será la función sigmoide logística, y softmax para la capa de salida. La inicialización de pesos de la red será siguiendo una distribución uniforme, con la varianza sugerida por Glorot & Bengio [1]. En el caso de la red CNN utilice función de activación ReLU y softmax para la capa de salida.

La red neuronal está configurada para detener su entrenamiento por una de las siguientes razones: Número máximo de épocas alcanzado (1000); 15 validaciones seguidas con aumento en el loss de validación. La última condición se conoce como “Early Stopping”, y su función es evitar que la red memorice la base de datos de entrenamiento perdiendo generalización (por sobre-ajuste). El objetivo de esta tarea es observar el efecto que tiene la elección de ciertos funcionales e hiperparámetros sobre el entrenamiento y el desempeño de un perceptrón multicapa. No se olvide de separar su base de datos en entrenamiento, validación y test.

Entre los parámetros de la red que usted debe analizar se tiene: el número de neuronas en la capa oculta, la tasa de aprendizaje, la función objetivo, el algoritmo de entrenamiento y el uso de regularizaciones. Para evaluar los resultados de la red se utilizarán matrices de confusión, curvas ROC y curvas DET (Detection Error Tradeoff)³, además de inspeccionar las curvas de aprendizaje. A continuación, se especifica lo que usted deberá desarrollar en esta tarea:

1. Para la construcción de su red MLP, utilice los hiperparámetros de la tabla:

Hiperparámetro	Valor
# neuronas en capa oculta	500
Función de costo	Entropía cruzada
Algoritmo de entrenamiento	SGD
Tasa de aprendizaje	0,1
Tamaño del mini-batch	5000
Regularización	-

Utilice el criterio de detención previamente mencionado. Luego reemplace la función objetivo por el error cuadrático medio y compare el aprendizaje y resultados obtenidos. Repita 5 veces sus experimentos para que sus comparaciones tengan validez y no sean fruto del azar en la inicialización. ¿Qué funcional es más adecuado para entrenar esta red? Justifique su respuesta en base a los resultados obtenidos y a los fundamentos teóricos.

2. Cree una red CNN con 2 capas de extracción (una capa convolucional seguida de una capa maxpooling) y 2 capas fully connected (FC) con los parámetros de entrenamiento, función de costo y batch. Realice 5 combinaciones de parámetros de tamaño de filtro, padding, stride y tamaño de neuronas en FC y compare sus desempeños. Elija la mejor combinación justificando su respuesta en base a los resultados obtenidos y a los fundamentos teóricos.
3. Con el funcional fijado en Cross Entropy, evalúe el impacto de cambiar la tasa de aprendizaje en ambas redes. Para ello compare cualitativa y cuantitativamente el entrenamiento y desempeño final del clasificador para los siguientes valores de tasa de aprendizaje: 10^{-2} , 10^{-1} , 10^0 y 10^1 . ¿Cómo cambian los tiempos de entrenamiento y tasa de error alcanzados? ¿Qué ocurre con la estabilidad del entrenamiento para tasas altas? Repita cada experimento 5 veces para considerar los efectos de la inicialización de la red. Acompañe sus cifras con los

³Las curvas ROC y DET solo utilizan 2 clases. Seleccione una de sus clases y compárela con el resto.

errores respectivos (desviación estándar sobre las 5 realizaciones). Se aconseja automatizar dichas repeticiones con un ciclo for y así evitar los procedimientos y cálculos tediosos a mano.

4. Considere la mejor tasa de aprendizaje encontrada en el punto anterior en su red MLP, es decir, con un buen compromiso entre rapidez, estabilidad y tasa de acierto final. Determine experimentalmente el número óptimo de unidades en la capa oculta (N) en función de los resultados obtenidos en el conjunto de validación (% de clasificaciones correctas). Entregue los resultados de porcentaje promedio de clasificaciones correctas según N . Experimente con 25, 100, 300 y 1000 neuronas, así como también eliminando la capa oculta (requiere modificar levemente el código). Modifique el criterio de early stopping, aumentando a 1000 las validaciones consecutivas permitidas que empeoran el loss, de forma tal que pueda observar el sobreajuste. Recuerde repetir 5 veces sus experimentos para que sus comparaciones tengan validez y no sean fruto del azar en la inicialización.
5. Con la tasa de aprendizaje ya elegida y el mejor número de neuronas ocultas, compare el algoritmo de gradiente descendente estocástico con ADAM en ambas redes. Utilice los parámetros por defecto que pytorch tiene para ADAM. Compare las mejores tasas de acierto encontradas por ambos algoritmos, tanto para entrenamiento como para validación. ¿Qué ventajas observa al utilizar el algoritmo ADAM? Recuerde ejecutar 5 veces.
6. Modifique la función de costos añadiendo una penalización por norma L2 de los pesos (a.k.a. weight decay) en ambas redes. Modifique el criterio de detención, cambiando el número de validaciones con error creciente de 15 a 1000. Analice y recuerde ejecutar 5 veces.

Muestre y explique el fenómeno de sobreajuste para el modelo no regularizado. ¿Afecta la regularización al sobreajuste? ¿Cómo impacta la penalización L2 al nivel de acierto alcanzado por la red? ¿Cómo cambia la distribución de los pesos en el caso regularizado?

7. Finalmente, en su informe deberá explicar el método de gradiente descendente estocástico con momentum: objetivo, funcionamiento, justificaciones, ventajas y desventajas. También explique el método ADAM y contrástelo con gradiente descendente con momentum. Además deberá comentar acerca de la elección del tamaño del mini-batch y su efecto en el entrenamiento.

Resultados e Informe

Se debe entregar un informe con los puntos solicitados anteriormente y el respectivo análisis de los resultados obtenidos. El análisis debe incluir comparaciones entre lo obtenido en la práctica y la teoría. Respalde su análisis con tablas y gráficos, procurando que sus figuras sean legibles, auto explicativas y pertinentes. Un exceso de gráficos sin un análisis adecuado será calificado negativamente.

- La tarea es individual.
- La entrega del informe (en formato PDF) y códigos es hasta el viernes 22 de Noviembre de 2024 hasta las 23:59:59 hrs. a través de CANVAS.
- Los atrasos se penalizarán con 0.5 puntos menos por día de atraso.
- Se recomienda que el informe no exceda las 10 páginas. En el caso que posea más graficas inclúyalas como anexos.
- La estructura del informe debe incluir:
 1. Introducción.
 2. Descripción concisa de los métodos y algoritmos usados.
 3. Resultados.
 4. Análisis y discusión de resultados.
 5. Conclusiones.
 6. Referencias.

Referencias

- [1] X. Glorot and Y. Bengio, “Understanding the difficulty of training deep feedforward neural networks,” in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256, 2010.