Tarea 1: MLP

Entrega: Viernes 1 de octubre, 23:59

Profesor: Pablo Estévez V.
Auxiliar: Ignacio Reyes J.
Semestre: Primavera 2021

La tarea consta de dos secciones, una teórica y otra práctica. Usted deberá entregar un informe con las respuestas de ambas partes. En la parte práctica debe ejecutar los experimentos pedidos, mostrar los resultados, realizar el análisis respectivo y las conclusiones respondiendo las preguntas presentes en el enunciado.

El informe debe ser conciso, evitando extenderse más allá de las 10 páginas (límite de páginas flexible).

Parte teórica

Responda las siguientes preguntas en no más de uno o dos párrafos por pregunta.

1. ¿Cuál es la ventaja del perceptrón multicapa respecto a la regresión logística? ¿Cuál es la importancia de contar con capas ocultas en el modelo? ¿Qué diferencias tienen las fronteras de decisión de dichos modelos?

2. ¿Qué efecto tiene el número de neuronas en la capa oculta del MLP sobre la capacidad del modelo? Explique el compromiso entre capacidad de un modelo y sobreajuste.

3. ¿Qué es la tasa de aprendizaje y cómo afecta el proceso de entrenamiento de una red neuronal? ¿Cómo se puede elegir la tasa de aprendizaje?

4. ¿Qué es un *mini-batch* y para qué sirve? ¿Cuál es la diferencia entre iteración y época?

5. Explique los conceptos de *accuracy*, *precision*, *recall* y *F1 score*.

Parte práctica

Esta tarea tiene por objetivo realizar un clasificador de dígitos manuscritos usando redes neuronales MLP. Para esto, se pide que entrene sus redes usando la base de datos MNIST, la cual contiene 70000 muestras de números manuscritos repartidos en entrenamiento, validación y prueba (test). El conjunto de entrenamiento tiene 55,000 ejemplos, el de validación 5,000 y el de prueba 10,000. Cada muestra de la base de datos (Figura 1) es un mapa de 28x28 píxeles (784 pixeles).

Para simplificar la tarea, se pide construir un clasificador de un dígito determinado versus el resto, con lo que el problema se reduce a un problema de detección. Dado el eventual desbalance de ejemplos por clase, se selecciona un subconjunto de los ejemplos de la clase "el resto". En consecuencia, el conjunto de entrenamiento del problema tiene 10778 ejemplos, el de validación tiene 924 y el de prueba 1948.

Usted deberá detectar el dígito "8". La salida deseada de la red neuronal será (0, 1) si la muestra presentada es un 8, y (1, 0) si la muestra es un dígito distinto de 8. El dígito 8 corresponderá a la clase "positiva", mientras que el resto de los dígitos integrarán la clase "negativa".

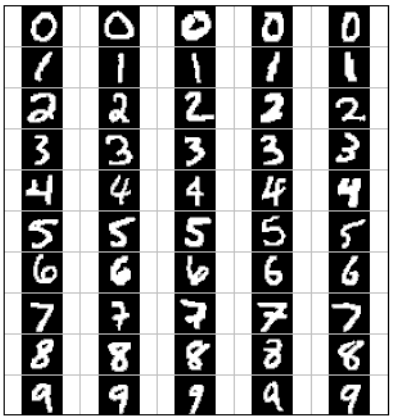

Figura 1: Muestras de la base de datos MNIST.

Se pide entrenar redes de arquitectura 784-N-2, donde N es el número de neuronas de la capa oculta. La función de activación 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 (2010).

La red neuronal está configurada para detener su entrenamiento por una de las siguientes razones: Número máximo de épocas[1] alcanzado (100); 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). Cada iteración de entrenamiento se realiza sobre un pequeño subconjunto de ejemplos (mini-batch) de tamaño 32. La validación se realiza cada 10 iteraciones de entrenamiento.

La red mencionada ya se encuentra implementada en los códigos que acompañan este enunciado. 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. Entre estos se cuentan el número de neuronas en la capa oculta, la tasa de aprendizaje y la función de costos.

[1] Una época corresponde a un recorrido completo por todas las muestras del conjunto de entrenamiento.

Para evaluar los resultados de la red se utilizarán matrices de confusión, curvas ROC y curvas DET (Detection Error Tradeoff), además de inspeccionar las curvas de aprendizaje.

Experimentos

Repita cada experimento 5 veces para considerar los efectos de la inicialización de la red. Acompañe sus cifras con los errores respectivos (desviación estándar sobre las 5 realizaciones). Las repeticiones se encuentran automatizadas en el código mediante un ciclo *for*.

Para el caso de las curvas de aprendizaje puede mostrar el resultado de una sola realización en la medida que las otras medidas muestren una baja sensibilidad ante las condiciones iniciales. También puede utilizar las curvas de aprendizaje para retratar casos anómalos, por ejemplo que falle la convergencia del algoritmo en algunas inicializaciones.

1. Ejecute en su totalidad el Jupyter Notebook "Tarea1.ipynb" utilizando los hiperparámetros de la tabla siguiente:

Hiperparámetro	**Valor**
# neuronas en capa oculta	25
Función de costo	Entropía cruzada
Algoritmo de entrenamiento	Backpropagation
Tasa de aprendizaje	0.1
Tamaño del mini-batch	32

   Utilice el criterio de detención previamente mencionado. Luego reemplace la función de costos por el error cuadrático medio y compare el aprendizaje y resultados obtenidos mediante las curvas de aprendizaje y tasas de acierto. ¿Hay alguna diferencia apreciable entre ambos casos? ¿Qué funcional es más adecuado para entrenar esta red? Justifique su respuesta en base a los resultados obtenidos y a los fundamentos teóricos.

2. Eligiendo la entropía cruzada como función de costos, evalúe el impacto de cambiar la tasa de aprendizaje. 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 cambia la tasa de error alcanzada al final del entrenamiento? ¿Qué ocurre con la estabilidad del entrenamiento para tasas altas? ¿Cuántas iteraciones se requieren para la convergencia del modelo en cada caso?. Justifique su respuesta apoyándose en las curvas de aprendizaje, tasas de acierto y otras métricas que considere pertinentes.

3. Considere la mejor tasa de aprendizaje encontrada en el punto anterior, 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). Experimente con 1, 10, 25 y 100 neuronas. Entregue los resultados de porcentaje promedio de clasificaciones correctas según N.

   Para el mejor valor de N, 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. No son necesarias las 5 inicializaciones para este caso.

4. Imagine que usted desea agrandar el conjunto de ejemplo positivos de la base de datos que posee. Para ello tiene a su disposición 1000 nuevos ejemplos no etiquetados. Los ejemplos que agregue a la base de datos deberán estar perfectamente etiquetados, por lo que se requiere que usted los inspeccione visualmente. Lamentablemente, revisar los 1000 ejemplos a mano le tomaría demasiado tiempo y sólo alcanza a chequear 400 casos en un tiempo prudente.

   Sin embargo, usted ya tiene un modelo de detección de dígitos manuscritos capaz de revisar estos ejemplos en un tiempo ínfimo, aunque con un nivel de desempeño no tan bueno como el suyo. Luego, podría utilizar el mejor modelo entrenado en las partes anteriores para que le asista en el etiquetado y le muestre los 400 casos más prometedores.

   ¿Qué impacto tiene mover el umbral de clasificación sobre la cantidad de casos predichos como positivos?. Elija un umbral tal que la cantidad de casos predichos como positivos sean aproximadamente 400 si a la red se le presentan 1000 ejemplos (asuma que estos 1000 ejemplos obedecen la misma distribución del conjunto de entrenamiento utilizado para construir el modelo). Indique cuál es dicho valor de umbral y el punto asociado en la curva ROC, explicando cómo llegó a esta elección. En este punto de operación, ¿cuál es la cantidad esperada de casos negativos que son erróneamente etiquetados por el modelo como positivos?

Referencias

- LeCun, Yann, et al. “Gradient-based learning applied to document recognition.” Proceedings of the IEEE 86.11 (1998): 2278-2324.
- Abadi, Martín, et al. “Tensorflow: Large-scale machine learning on heterogeneous distributed systems.” arXiv preprint arXiv:1603.04467 (2016).
- Glorot, Xavier, and Yoshua Bengio. “Understanding the difficulty of training deep feedforward neural networks.” Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics. 2010.