UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

Informe: Tarea #1

Informe sobre arquitecturas modernas de Redes Neuronales profundas
Universidad Central de Venezuela por el
Estudiante. Rafael Eduardo Contreras.

Profesor: Fernando Crema

Caracas, jul 15 de 2025

UNIVERSIDAD CENTRAL DE VENEZUELA
FACULTAD DE CIENCIAS
ESCUELA DE COMPUTACIÓN

ÍNDICE GENERAL

Notebooks

Notebook 3.4 - Activation functions - Link

Considero que este fue muy sencillo, la mayoría de los ejercicios era solo definir funciones. Sobre como presentan el contenido me gusto bastante la forma en que grafican la red neuronal, mostrando como cada función lineal se suma con las otras.

Uno de los puntos importantes en este notebook era aprender porque las funciones de activación no son lineales, esto ya lo vimos en ML. Aun así bueno recordarlo y ver como lo explican aquí.

Mencionan que es mejor usar otras funciones y no la sigmoide. Asumo esto es porque usar ReLU es mucho más rápido en el paso hacia atrás.

Notebook 4.1 - Composing networks - Link

Este me gusto bastante, fue divertido hacer la red “esperada”. Luego de agarrar que el rango de la primera es el dominio de la segunda (lógicamente), se hace bastante sencillo.

Realmente lo vi como un trabajo bastante repetitivo luego de agarrarle los pasos a la composición. Finalmente, con la reflexión final me hizo darme cuenta de que una composición de dos funciones tendrá la misma cantidad de regiones lineales que el producto de la cantidad de regiones de ambas funciones.

Notebook 4.2 - Clipping functions - Link

Este costo más que los anteriores. Fue más que todo seguir las fórmulas del libro. Aplicaron la misma técnica de enseñanza que vimos anteriormente de graficar cada función por la que pasa la red y luego ver como su suma crea otra función.

La sección final fue bastante directa, cambiar valores y pensar en como cambiaria la función final. Muestra como el valor de los pesos afecta el resultado final.

Notebook 4.3 - Deep networks - Link

Este costo más que los anteriores, unos cuantos intentos para obtener las redes correctas. Bastante visual saber que tenemos el resultado correcto.

Ya conocía bastante sobre como se calcula la inferencia en una NN, así que no fue muy revelador aprender como se convierten estas funciones en operaciones con matrices.

Notebook 5.1 - Least squares loss - Link

En este los comentarios fueron hechos con la anotación "Res" y "Edit".

El inicio fue bastante sencillo, el resultado de cambiar la media y varianza en una distribución normal se aprende en EyP. Continuando con el "negative log likelihood" es interesante y evidente la relación entre el mínimo valor de esta función y el máximo del likelihood.

También quiero destacar como muestran la gráfica donde relaciona el likelihood con el cambio de la variable beta. Me surge la duda de porque se logró llegar al valor máximo del mismo solo

cambiando esta variable sin variar las otras. Capaz este problema fue hecho de tal manera que el máximo se conseguía solo al variar beta.

Notebook 5.2 - Binary cross-entropy loss - Link

Estos temas ya lo tratamos durante ML. Igual fue divertido hacer las funciones dadas las fórmulas del libro.

Se hace lo mismo que en el notebook anterior, viendo como llegamos al mejor resultado solo variando un parámetro de la red. Aquí no hicimos comentarios en el código, capaz porque fueron muy directos los resultados.

Notebook 5.3 - Multiclass cross-entropy loss - Link

Aquí nos introducen por primera vez la función softmax (ya vista en ML). Una de las implementaciones no me estaba funcionando y al correr el notebook denuevo, funciono, un comentario lo expresa como:

> #rafael: pienso esto esta incorrecto. Ya que no me daban los valores abajo

De resto es muy similar a los últimos dos notebooks. Mostrándonos como cambia nuestra clasificación, error y likelihood al cambiar beta1.

Notebook 6.2 - Gradient descent - Link

Revise varias veces este notebook y el siguiente debido a que en el experimento tenía bugs relacionados con el paso.

Me parece algo confuso como actualizan los parámetros dado el gradiente, me refiero a la notación. Luego, al leer el código, entendí mejor como eran los pasos (igual ya lo habíamos visto en ML, había que refrescar).

Me parece extraño que la función que grafican como ejemplo no avance más rápido. En las primeras 3-4 iteraciones hace más trabajo que en el resto. De resto el notebook es bastante sencillo de resolver, no tiene ejercicios tan complejos.

Notebook 6.3 - Stochastic gradient descent - Link

Al igual que el anterior, revise varias veces este notebook a causa de bugs que tenía en mi experimento. La estructura es similar al anterior y los ejercicios no son difíciles.

Deje comentarios con la nota "rafael", entre ellos los otros valores que use para que el resultado cayera en mínimos locales. Me gusta la manera en la que muestran como cambia la función durante el aprendizaje.

En la sección donde preguntan que ocurre con un alpha muy pequeño y muy grande vemos de forma experimental algo que ya fue discutido en clases. Esto es que saltos muy amplios se "salen" del valle y saltos muy pequeños tardan en llegar al mínimo.

Me ayudo durante el experimento la diferenciación de las funciones en el SGD y el GD.

Notebook 6.4 - Momentum - Link

Desde un inicio me intereso bastante el concepto de momentum, me preguntaba como se define matemáticamente (esto fue, durante la clase), imagine sería necesario alguna variable indicando la dirección del paso anterior.

Fue bastante sencillo el ejercicio que mostraron, simplemente era necesario usar la fórmula del libro para el paso con momentum. Sobre el momentum de Nesterov todavía me falta comprender más la fórmula del mismo, entiendo usa el momentum previo para cambiar el gradiente que se usa para el momentum actual (y paso actual).

En el caso de la función en el ejercicio, un momentum "fuerte" (0.9) ayudo bastante a llegar al mínimo más rápido, me pregunto si esto ocurrirá al aproximar otras funciones.

Notebook 6.5 - Adam - Link

Me parece curiosa la idea de Adam de hacer una actualización constante en todas las dimensiones, independientemente del valor de la pendiente. Me gustaría saber por qué termina siendo más efectivo que al tomar en cuenta el nivel de pendiente por dimensión.

De resto es bastante sencillo ver en la fórmula como solo toma la dirección (+ o -), al elevar al cuadrado el denominador y sacar su raíz aseguramos que sea positivo, tomando el signo del numerador. De resto es igual al momentum que veíamos antes.

Notebook 7.1 - Backpropagation in toy model - Link

Este notebook me ayudo mucho más a entender el paso hacia atrás. De hecho, en mi experimento, use la misma función para crear el dataset inicial para probar que la red neuronal estaba funcionando.

Durante la creación del experimento revisando estos notebooks todos estos temas hicieron click en conjunto. Usamos el backprop para calcular las derivadas que luego (según el optimizador) usaremos para actualizar los pesos. Las derivadas son con base en la función de perdida para poder minimizarla.

Este notebook fue mucho más sencillo que el siguiente, ya que el cálculo de derivadas fue a mano prácticamente, no programáticamente. Ayudo bastante que el autor hiciera las primeras derivadas en cada paso. Las comprobaciones finales son muy importantes, perfectamente se puede hacer todo de manera equivocada si estas no se incluyen.

Notebook 7.2 - Backpropagation - Link

Muy útil al momento de hacer el experimento. El algoritmo es el mismo, solo que varios pasos se hacen en módulos distintos en mi código. Ayudo bastante a debuggear mi experimento.

La sección del forward pass es bastante sencilla de hacer. En la del backpropagation necesité las fórmulas del libro, bastante. El bug más frecuente se relacionaba a dimensiones incorrectas durante operaciones entre NDArrays.

Al final me di cuenta no use la función indicador, preferí usar np.where en numpy. Al igual que antes, es excelente tener manera de comprobar los resultados.

Notebook 7.3 - Initialization - Link

Todavía me falta por entender por qué no explotan los valores. También todavía me falta entender la inicialización para el forward pass y backward pass. Comprendo la fórmula, me falta entender de donde viene.

De resto, los ejercicios del notebook me parecieron sencillos. Pienso es más fácil entender el tema usando el libro más que el notebook, ya que esta muestra las formulas y busca explicarlas. A nivel práctico igual es bueno ver como explotan los valores o desaparecen.