Examen Parcial N° 1

Entrega : a partir del miércoles 5 de mayo hasta el domingo 9 de mayo.

Modalidad

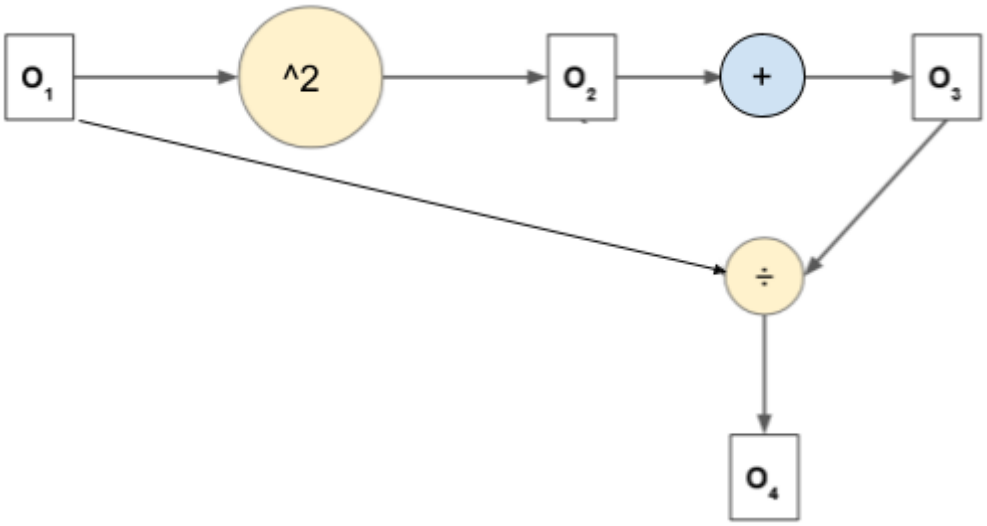
- El examen tiene 4 secciones de ejercicios.
- Elija al menos 5 ejercicios de cada sección.
- Nota relativa Q1, Q2, Q3, Q4 según puntajes del curso.
- Puede usar bibliografía
- La presentación y evaluación será individual.
- Cree un documento pdf con las pregunta que eligió y respóndalas a continuación
- Puede resolver los ejercicios a mano si es conveniente e insertarlas al documento.

Sección 1: MLP y modelos lineales

1. Si tenemos un dataset con 200 ejemplos y 9 atributos (features) y tenemos 8 clases. ¿Qué tamaño tiene el tensor de pesos? ¿Qué tamaño tiene el tensor de bias? ¿ Qué tamaño tiene el tensor de diseño? ¿Qué tamaño tiene el tensor de etiquetas?
2. Hemos visto 3 versiones de descenso de gradiente. Explíquelas. (Hablamos de variantes de descenso de gradiente, no de algoritmos de optimización)
3. Comente línea por línea qué hace este código:

```
num_epochs = 3
for epoch in range(num_epoch):
      for X, y in data_iter:
            l = loss(net(X), y)
            trainer.zero_grad()
            l.backward()
            trainer.step()
```

4. Se quiere introducir un modelo MLP para detectar diabetes. Los médicos clasifican en Diabetes tipo 1, diabetes tipo 2, en riesgo de contraer diabetes y fuera de peligro. ¿Qué función de pérdida usaría para entrenar a este modelo? Justifique
5. Observe el siguiente grafo computacional:



   En amarillo se indican operaciones que actúan de componente a componente; en celeste operaciones entre componentes. Si O_1 es un vector, ¿qué es O_3? ¿Qué obtenemos al final en O_4?
6. ¿Por qué no es una buena idea iniciar los pesos del modelo en 0 (cero)?
7. ¿Qué diferencia hay entre parámetros e hiperparámetros?
8. Observe la siguiente fórmula;

$$\partial_{W1} \text{Loss} = \partial_o \text{Loss} \cdot \partial_h o \cdot \partial_y h \cdot \partial_{W1} y$$

   Si este es el gradiente para los pesos de un MLP, ¿Puedo cambiar de orden las multiplicaciones? (Recuerde que ∂_{W1} es una derivada respecto a una matriz)

9. Compare las siguientes funciones:

$$f(x) = \max(x,0)$$

$$g(x) = xe^{sx}/(1 + e^{sx})$$

¿Qué ocurre en g(x), si el parámetro s tiende a infinito? ¿Qué función es g si s = 1?

10. Considere las siguientes funciones:

$$f(x,y,z) = \max(x,y,z)$$

$$g(x,y,z) = xe^{sx}/(e^{sx} + e^{sy} + e^{sz}) +$$

$$+ ye^{sy}/(e^{sx} + e^{sy} + e^{sz}) +$$

$$+ ze^{sz}/(e^{sx} + e^{sy} + e^{sz})$$

¿Qué ocurre si el parámetro s tiende a infinito? ¿Qué relación encuentra con SoftMax, Max y la función sigmoidea?

11. La función sigmoidea puede escribirse como:

$$s(x) = e^{x} / (e^{x} + 1)$$

Por otro lado la tangente hiperbólica puede escribirse como:

$$\tanh(x) = (e^{x} - e^{-x})/(e^{x} + e^{-x})$$

Encuentre un cambio de variable t = ax tal que

$$s(x) = c*\tanh(t) + d$$

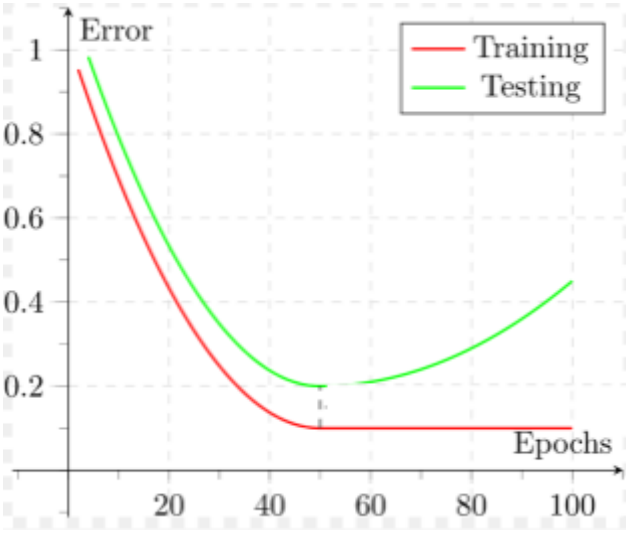
12. ¿Por qué es necesaria siempre una función de activación?
Supongamos que tenemos una función de activación leaky ReLU. ¿Qué valor de hiperparámetro k vuelve inútil a leaky ReLU?¿Por qué?

$$\text{leakyReLU}(x) = \max(x,0) + k*\min(x,0)$$

13. Dados dos conjuntos convexos, su intersección es convexa. Demuestre esta afirmación a partir de la definición de conjunto convexo.

Sección 2: Implementación, estadista y optimización

1. La API nn.Module de PyTorch permite generar modelos definiendo solo 2 funciones. Nombre cuáles son y para qué sirve cada una.
2. ¿Por cuáles motivos uno desearía guardar en un archivo los parámetros de nuestros modelos? ¿Hay motivos para hacerlo, aún cuando no hemos terminado el entrenamiento?
3. Defina con sus palabras el error de entrenamiento y el error de generalización.
4. Explique las 3 componentes del error de generalización.
5. Analice el siguiente gráfico:



Esta es la función de pérdida de un modelo. En rojo los datos de entrenamiento, en verde los datos de testeo. El modelo final tras las 100 épocas, ¿tendrá un rendimiento aceptable? Justifique. Si el rendimiento no fuera aceptable, ¿que podría haber hecho para que los sea?

6. ¿Usted implementaría dropout durante la predicción?
7. Considere una función de pérdida con regularización L2

   $$f(\mathbf{W},\mathbf{b},k) = loss(\mathbf{W},\mathbf{b}) - kL_2(\mathbf{W},\mathbf{b})$$

   ¿Qué efecto tiene un k mayor durante el entrenamiento?
8. ¿Qué ocurre si la probabilidad de dropout es 1? ¿y si fuera 0?
9. ¿Qué problemas tiene el cálculo del Hessiano para nuestros modelos? Analice tanto gasto de memoria y el costo computacional.

10. Una con flechas

Momentum	usa media móvil
adagrad	usa la varianza para aproximar hessiano
RMSPRop	el learning rate es una media móvil
Adadelta	
ADAM	

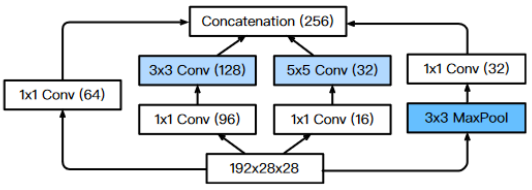
11. Dada una serie de datos con 1000 datos:

    Si los monomios de potencias distintas, los exponentes de bases distintas y los senos y cosenos con diferente frecuencias son funciones linealmente independientes entre ellas, ¿cuántas funciones linealmente independientes son necesarias para un ajuste perfecto?
    Considerando lo que sabemos de underfitting y overfitting, ¿Qué espera que pase para estos casos?

12. Suponga que accidentalmente duplica los datos de su dataset. ¿Cómo respondería cada variante de descenso de gradiente?

Sección 3: Convoluciones y redes convolucionales

1. ¿Que condiciones para poder trabajar con imágenes cumplen las convoluciones? Explíquelas
2. En un modelo se aplican 2 capas convolucionales. Cada kernel es de 3 x 3. ¿Qué tamaño tiene el campo receptivo de los píxeles del mapa de atributos generados por la segunda capa?
3. ¿Qué parámetros tiene una capa de pooling?
4. A la entrada de una capa de pooling hay 20 canales.¿Cuántos canales tendrá a la salida? ¿Y si hubiera 500 a la entrada? ¿y con 5 a la entrada?
5. ¿Qué diferencia hay entre los enfoques para visión computacional pre image-net y las redes que ganaron popularidad tras image-net?
6. ¿Por qué las capas densas, al estilo de los perceptrones, se encuentran al final de una CNN y no entre medio de capas convolucionales?
7. Explique porqué las convoluciones con kernel 1 x 1 se comportan como una capa densa a nivel de píxel.
8. ¿Por qué un bloque inception es más eficiente que una capa convolucional de 5 x 5? Suponga que en implementaciones, la cantidad de canales de entrada es 192 y la cantidad de canales de salida es 256. Justifique con el cálculo del gasto computacional y de memoria.



ejemplo de bloque incepction

9. ¿Qué precauciones deben tomarse para poder aplicar batch normalization durante la predicción?
10. ¿Cuántos parámetros tiene una capa de normalización por lotes en una capa convolucional, si la entrada tiene forma 28 x 28 x 120?
11. ¿Cuál es la ventaja de que un bloque residual pueda convertirse rápidamente en una identidad? Considerar el concepto de funciones anidadas
12. ¿Por qué los bloques residuales y densos ayudan a evitar el desvanecimiento de gradiente?
13. ¿Por qué los bloques densos son más eficientes que los residuales?¿Por qué son más expresivos?
14. Definida una imagen de 100 x 300 con un padding de 5, una convolución 7 x 9 y un stride 8, ¿cuál es el tamaño de la imagen de salida?¿Cuales el número de multiplicaciones necesarias?
    Si cada número es un byte en memoria, ¿cuánta memoria necesitamos para implementar la capa?
15. Un truco muy común para disminuir el ruido consiste en hacer integrales de valores. Piense en cómo implementaría una integral numérica. ¿Es necesario implementar esto en CNN? Justifique

Sección 4: Aplicaciones de redes convolucionales

1. Explique el concepto de Fine Tuning.
2. Cuando hacemos transferencia de estilo, ¿cuáles son los parámetros entrenables?
3. En detección de objetos, nuestro problema consta de dos partes. Identificar al objeto y delimitarlo con un rectángulo. Nuestra función de pérdida tiene al menos dos términos ¿Qué término usamos para identificar y delimitar?
4. ¿Qué valores puede tomar la operación, intersección sobre unión? ¿A qué corresponden el máximo y el mínimo?
5. ¿La convolución transpuesta, en general, es lo mismo que una deconvolución? Justifique
6. Las convoluciones transpuestas en PyTorch reciben dos parámetros llamados padding y stride. ¿A qué hacen referencia estos valores?
7. Considere una Fully Convolutional Network usada para segmentación semántica. ¿Cómo decide cuál es la categoría que le corresponde a cada píxel, dado que la salida de la red es (cantidad de clase + 1, alto, ancho)?
8. Quiero que la salida de mi convolución transpuesta sea cuatro veces más grande que la entrada. ¿Qué valores de stride, padding y tamaño de kernel se requieren?
9. ¿Qué diferencia hay entre el aprendizaje generativo y el aprendizaje discriminativo?
10. ¿Por qué decimos que el generador y el discriminador son redes adversarias? Justifique con las funciones de pérdida
11. Considere una convolución con kernel de tamaño 3 x 3. Supongamos que tenemos una imagen de entrada 5 x 5. Llamemos a la salida O.
    ¿Qué tamaño tiene O?
    ¿Existe otra entrada 5 x 5 que también nos devuelva la misma O?
    En función de lo anterior y de las definiciones de función inyectiva y función sobreyectiva, ¿es la convolución una función inyectiva? ¿es sobreyectiva?
12. Considere una convolución transpuesta con kernel K. Supongamos que tenemos una imagen de entrada 5 x 5. Llamemos a la salida P. P es 10 x 10
    ¿Existe otra entrada 5 x 5 que también nos devuelva la misma P?
    En función de lo anterior y de las definiciones de función inyectiva y función sobreyectiva, ¿es la convolución una función inyectiva? ¿es sobreyectiva?