Universidad Técnica Federico Santa María
Departamento de Informática

INF-477 Redes Neuronales Artificiales
Cuestionario I - 2019.

1. ¿Cuando recomendaría el uso de una red neuronal profunda en vez de un modelo clásico de aprendizaje automático?
2. Dibuje una red neuronal con neuronas MC (McCulloch & Pitts) que compute la función XOR. Use la notación vista en clases.
3. Demuestre que una red neuronal puede implementar cualquier función booleana de d variables.
4. ¿A qué nos referimos con "profundidad" en una red neuronal? ¿Porqué este término se ha vuelto tan relevante en los últimos años?
5. Diseñe una red neuronal con a lo más 1 capa escondida que compute la función de paridad de d variables booleanas $x_1, x_2, \ldots, x_d$.
6. Diseñe una red neuronal con a lo más $3(d-1)$ neuronas que compute la función de paridad de d variables booleanas $x_1, x_2, \ldots, x_d$. ¿Es posible construir una red de 1 sola capa que tenga un número similar $\mathcal{O}(d)$ de neuronas?
7. Es cierto o es falso que una red neuronal de 1 sola capa escondida puede implementar cualquier función $f : \mathbb{R}^d \to \mathbb{R}$ con "precisión arbitraria"? ¿Qué significa esto último?
8. ¿Cuál es el rol de las capas ocultas de una red neuronal?
9. ¿Cuál es la diferencia entre los algoritmos que hemos denominado *Perceptron* y *Regla Delta* al describir la historia de los desarrollos que condujeron a las redes neuronales modernas?
10. Considere una red neuronal con topología arbitraria, representada en forma de un digrafo de computación. Suponga que es posible identificar un conjunto de nodos de entrada, un conjunto de nodos de salida, y que no existen ciclos en la red. ¿Es cierto o es falso que esta red se puede siempre estructurar como red de tipo feed-forward? ¿Depende la complejidad de un *forward pass* (cálculo de la salida a partir de la entrada) del grado de conectividad de cada neurona? ¿Depende la complejidad de un *backward pass* (cálculo de la señal de error correspondiente a 1 peso de una neurona) del grado de conectividad de cada neurona?
11. Considere una red neuronal feed-forward estándar con capas completamente conectadas y arquitectura $1024 \times 100 \times 100 \times 10$ (capa de entrada de 1024 neuronas, capa de salida de 10 neuronas, y 2 capas escondidas de 100 neuronas). Determine el número total de parámetros entrenables de esta red. Indique los supuestos realizados.
12. Suponga que debe entrenar una red neuronal (feedforward estándar) para clasificar opiniones (textos breves) entre *positivas* y *negativas*. Para ello, dispone de un conjunto de n textos etiquetados $D = \{(d^{(\ell)}, y^{(\ell)})\}_{\ell=1}^{n}$ donde $d^{(\ell)}$ corresponde a una secuencia/lista de palabras, pertenecientes a un vocabulario $\mathcal{V}$, e $y^{(\ell)} \in \{+1, -1\}$ es la categoría asignada por un grupo de humanos. Explique: (i) ¿Cuántas neuronas salida tendría la red? (ii) ¿Qué tipo de función de activación utilizaría en la capa de salida?, (iii) ¿Qué función de pérdida (loss) seleccionaría para el entrenamiento? (iv) ¿Cómo preprocesaría los textos para que los pueda leer la red?, (iv) ¿Cómo elegiría el número de capas ocultas y el número de neuronas?

13. Suponga que debe entrenar una red neuronal (feedforward estándar) para estimar la concentración de una cierta sustancia química a partir de una imagen tomada justo después de una determinada reacción. Para ello, se dispone de un conjunto de n ejemplos $D = \{(m^{(\ell)}, y^{(\ell)})\}_{\ell=1}^{n}$ donde $m^{(\ell)}$ corresponde a una imagen de $128 \times 128 \times 3$, e $y^{(\ell)} \in \mathbb{R}$ es el valor real de la concentración de la sustancia. Explique: (i) ¿Cuántas neuronas de entrada y salida tendría la red? (ii) ¿Qué tipo de función de activación utilizaría en la capa de salida?, (iii) ¿Qué función de pérdida (loss) seleccionaría para el entrenamiento?

14. Considere una capa densa ("fully connected") tradicional $y = \sigma(Wx + b)$. Si $x \in \mathbb{R}^{100}$ e $y \in \mathbb{R}^{10}$, ¿Cuál es la forma de las matrices W y b? ¿Cuál es la forma de la matriz $\partial y / \partial x$?

15. Suponga que tenemos una red con 3 capas ocultas densas como en la pregunta anterior. Escriba la ecuación de la capa 3 si queremos agregarle "skip connections" desde la capa de entrada.

16. Explique la ventaja más importante de utilizar funciones de activación *ReLu* $s(\xi) = \max(0, \xi)$ en vez de funciones de tipo "squashing" en el diseño de redes neuronales.

17. Explique brevemente el problema denominado "desvanecimiento de los gradientes" en redes neuronales profundas. Mencione dos "innovaciones" de la última década que hayan contribuido a atenuar este problema, haciendo posible el entrenamiento de redes muy profundas.

18. ¿Por qué los pesos de una red suelen inicializarse muestreando una distribución de probabilidad? ¿Cuál es el criterio utilizado en la práctica para elegir el valor esperado de la distribución? ¿Cuál es el criterio utilizado en la práctica para elegir la varianza de la distribución?

19. Considere el grafo dirigido de computación definido en la Figura 1. Suponga que el estado de un nodo v se obtiene multiplicando el estado de los nodos incidentes $\mathrm{Pa}(v)$ por los correspondientes peso de conexión, y sumando estos resultados. Escriba un conjunto de ecuaciones que permitan calcular $\partial E/\partial p$, $\partial E/\partial r$, $\partial E/\partial q$, $\partial E/\partial u$, $\partial E/\partial v$, $\partial E/\partial w$, $\partial E/\partial x$, e $\partial E/\partial y$, a partir de $\partial E/\partial s$, usando un número de sumas y productos que no exceda $2A$, donde A es el número total de arcos del grafo.

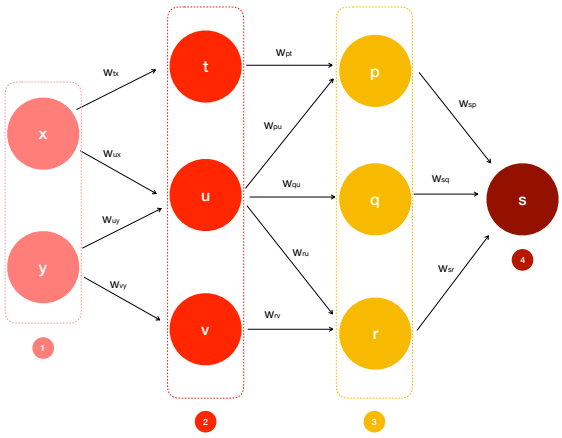

Figure 1: Grafo para la pregunta 19.

20. Explique el significado de parámetro *"tamaño de batch"* (*batch size*) en la implementación moderna del algoritmo BP (back-propagation). ¿Que valor recomendaría si su conjunto de entrenamiento es de 10.000 ejemplos? ¿Que valor recomendaría si su conjunto de entrenamiento es de 1.000.000 de ejemplos?

21. ¿Qué es una "epoch" en el entrenamiento de redes neuronales?

22. Demuestre que la función de pérdida (loss) denominada "cross-entropy" $L(y, f) = -y \log f - (1 - y) \log(1 - f)$, $y, f \in [0, 1]$ es una función convexa de f. Explique después porqué el entrenamiento de una red neuronal con esta función de pérdida da origen a una función no-convexa de los parámetros de la red.

23. Demuestre que si tenemos infinitos ejemplos de entrenamiento, disponemos de un algoritmo de entrenamiento que garantiza convergencia al óptimo global y entrenamos una red neuronal con la función de error cuadrática $L(y, f) = (y - f)^2$, la red terminará implementando la función de regresión $E[y|x]$.

24. Explique qué es la denominada "tilted loss" y para que podría servir entrenar una red neuronal con esta función de pérdida.

25. Explique en qué consiste la técnica denominada *progressive decay* para el entrenamiento de redes neuronales artificiales. ¿Cuál es su principal desventaja?

26. Explique qué es un punto silla y de un argumento de porqué este tipo de puntos son mucho más relevantes que los denominados "mínimos locales" en el entrenamiento de redes neuronales modernas. Mencione tres técnicas que se usen para permitir "escapar" de estos puntos al entrenar una red con back-propagation.

27. Considere la función $J(x, y) = 4x^2 + y^2$. Muestre las primeras 4 iteraciones del método de optimización que hemos denominado "gradiente descendente, partiendo de $(2, 2)$ y usando una tasa de aprendizaje fija e igual a 0.5.

28. Explique la diferencia y el objetivo de las técnicas que hemos denominado *momentum* y *momentum de Nesterov* para entrenamiento de redes neuronales artificiales.

29. Explique brevemente la diferencia entre las técnicas que hemos denominado *Adagrad*, *RMS-Prop* y *Adam* para entrenamiento de redes neuronales artificiales.

30. Demuestre el denominado "Lema de Descenso": si $J(\theta)$ es diferenciable y "suave", en el sentido de que satisface

$$\|\nabla J(\theta_1) - \nabla J(\theta_2)\| \leq L\|\theta_1 - \theta_2\| \,. \tag{1}$$

para algún valor de $L < \infty$. Entonces,

$$J(\theta_2) \leq J(\theta_1) + \nabla J(\theta_1)^T(\theta_2 - \theta_1) + \frac{L}{2}\|\theta_2 - \theta_1\|^2 \,, \tag{2}$$

31. Hemos dicho que un algoritmo de entrenamiento iterativo basado en gradiente que intenta minimizar una cierta función objetivo $J(\theta)$ "converge a tasa $\mathcal{O}(t)$" si después de $\mathcal{O}(t)$ iteraciones puede asegurar que

$$\min_{k=1,\ldots,t} \|\nabla J(\theta^{(k)})\|^2 < \epsilon \,, \tag{3}$$

¿Es cierto o es falso que la técnica denominada *momentum de Nesterov* permite mejorar la tasa de convergencia de SGD de $\mathcal{O}(t)$ a $\mathcal{O}(\sqrt{t})$?

32. Suponga que se entrena una red neuronal $f(x; \theta)$ con cierta función objetivo $J(\theta)$ diferenciable y "suave", en el sentido de que satisface la ecuación (1). Demuestre que si se emplea gradiente descendente (gradient descent) con tasa de aprendizaje fija η y esta satisface la condición $\eta \leq 2/L$, entonces se puede garantizar la convergencia a un punto estacionario.

33. Suponga que se entrena una red neuronal $f(x; \theta)$ con cierta función objetivo $J(\theta)$ diferenciable y "suave", en el sentido de que satisface la ecuación (1). ¿Es cierto o es falso que la técnica denominada *Adagrad* permite garantizar una tasa de convergencia de orden $\mathcal{O}(t)$ aún si la tasa de aprendizaje inicial no satisface la condición clásica $\eta \leq 2/L$?

34. Suponga que se entrena una red neuronal $f(x; \theta)$ con cierta función objetivo $J(\theta)$ diferenciable y "suave", en el sentido de que satisface la ecuación (1). Demuestre que si se emplea SGD con tasa de aprendizaje dinámica que satisface las denominadas "ecuaciones de Robbins-Monro", entonces se puede garantizar la convergencia a un punto estacionario "en valor esperado".