Universidad Técnica Federico Santa María
Departamento de Informática

INF-395 Introducción a las Redes Neuronales Artificiales
Certamen I-2017

Instrucciones:

- Este certamen debe ser resuelto individualmente.
- Puede usar apuntes, pero no puede intercambiar respuestas con otros estudiantes.
- Las respuestas se deben entregar en formato electrónico utilizando un editor de textos. Respuestas entregadas en latex tienen 20 puntos de bonificación.
- Si denotamos por p_i el puntaje obtenido en la pregunta i y por q_i al puntaje máximo de esa pregunta. La nota se calculará como

$$N = 100 \cdot \frac{\sum_i p_i}{\sum_i q_i} \ .$$

- Escriba explícitamente cualquier supuesto que crea importante y todos los pasos intermedios que sean necesarios para llegar a un resultado.
- Las preguntas con el símbolo $\star\star$ son opcionales para estudiantes de pre-grado, pero obligatorias para estudiantes de postgrado.
- Las preguntas con el símbolo †† son opcionales para estudiantes de postgrado, pero obligatorias para estudiantes de pre-grado.

Preguntas:

1. †† (10 puntos) Explique qué se entiende por "sobreajuste" (*overfitting*) en aprendizaje automático y cuándo es más común que se manifieste. Mencione dos técnicas específicas de que usaría para evitar este problema en el caso de redes neuronales artificiales.
2. †† (10 puntos) Explique cómo elegiría en la práctica el número de neuronas N_h y la tasa de aprendizaje (*learning rate*) η a utilizar en el entrenamiento de un MLP estándar. Suponga que dispone de un conjunto de n datos para esta tarea y que se entrena con tasa de aprendizaje (η) fija.. ¿Cambia su elección si n es muy muy grande con respecto al caso en que n es relativamente pequeño?
3. (10 puntos) Cite dos motivos por los cuales cuando se entrena una red neuronal repetidamente, manteniendo los parámetros de entrenamiento fijos, obtiene con frecuencia un resultado diferente.
4. (10 puntos) ¿Es correcto entrenar una red feed-forward para un problema de clasificación utilizando la función de pérdida denominada *binary cross entropy*? Justifique.
5. (20 puntos) Explique el problema del "desvanecimiento" o "explosión" del gradiente descendente en redes neuronales profundas. ¿Porqué este fenómeno es particularmente marcado en el caso de redes neuronales recurrentes? ¿Depende este fenómeno del tipo de función de activación que se utilice? Si su respuesta a la última pregunta es negativa, de un ejemplo.
6. (15 puntos) ¿Por qué se afirma que Dropout actúa como Bagging con un número exponencialmente grande de predictores[1]? ¿Por qué Dropout resulta menos costoso que un Bagging explícito con un gran número de predictores?

[1] Exponencialmente grande en el número de neuronas

7. †† (20 puntos) Suponga que debe resolver un problema de clasificación en el que se desea etiquetar imágenes de caras (humanas) de acuerdo a un conjunto predefinido de $K = 10$ "emociones" (alegría, sorpresa, etc) utilizando una red neuronal convolucional de varias capas ($L > 3$). Para ello, se ha recolectado un pequeño dataset de 32 imágenes para cada una de las emociones posibles. Explique:

   (a) ¿Qué estrategias utilizaría para "expandir" el conjunto entrenamiento disponible y porqué esto sería deseable? (Suponga que no puede recolectar más imágenes).

   (b) ¿Porqué tendría sentido o no pre-entrenar la red utilizando pre-entrenamiento no-supervisado por capas (unsupervised layer-wise pretraining)?

   (c) ¿Porqué tendría sentido o no pre-entrenar la red utilizando un gran dataset etiquetado de imágenes de otros objetos (entre los que se incluyen por ejemplo aviones, autos, personas y animales)?

8. (25 puntos) La función de pérdida denominada *cross-entropy* es una de las elecciones más comunes en el entrenamiento de redes neuronales artificiales modernas. Derive esta función de pérdida como caso especial de una estimación de máxima verosimilitud. Discuta además si esta elección resulta más apropiada para un problema de clasificación o de regresión y su relación con la teoría de la información.

9. †† (15 puntos) Explique la diferencia entre el algoritmo CD (Contrastive Divergence) y PCD (Persistent CD) para entrenar RBM's. Explique además qué denota CD_k.

10. (20 puntos) Considere una red neuronal convolucional alimentada con imágenes RGB de 32×32 pixeles. Suponga que a la capa de entrada sigue un capa convolucional con 16 filtros de 4×4, una capa de pooling con filtros de 2×2, otra capa convolucional con 32 filtros de 4×4, otra capa de pooling con filtros de 2×2 y finalmente un MLP con 256 neuronas ocultas y 10 neuronas de salida. Ilustre las transformaciones que sufre un patrón de entrada al pasar por cada capa durante el *Forward Pass*. Determine además el número total de parámetros de la red. Suponga que en ninguna de las capas se implementa padding.

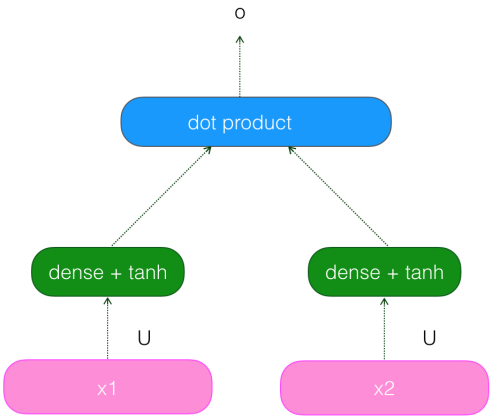

Figure 1: Figura para problema 11.

11. ⋆⋆ (25 puntos) Suponga que debe entrenar una red neuronal sobre pares de objetos (representados como vectores en $\mathbb{R}^d$) siguiendo la arquitectura de la Figura (1), donde la capa de salida (azul) efectua un sencillo producto punto entre los vectores entrantes y las capas ocultas (verdes) tienen pesos compartidos. El objetivo de la red es aprender a reconocer cuando dos objetos son similares o disimilares. Para ello se dispone de un dataset de pares $\{(x_1^{(\ell)}, x_2^{(\ell)})\}_{\ell=1}^n$. Escriba las ecuaciones correspondientes al backpropagation que usaría para entrenar esta red. Luego, extienda su algoritmo al caso de dos capas escondidas (con pesos compartidos).

12. †† (15 puntos) Suponga que dese entrenar una red neuronal para reconocer lenguaje de señas sobre videos. Para simplificar suponga que la red recibe una secuencia de frames de video correspondientes a

1 de K posibles gestos y que se tiene un abundante conjunto de entrenamiento. Explique qué tipo de arquitectura consideraría.

13. ⋆⋆ (25 puntos) Considere una RBM con estados visibles $\mathbf{v} = (v_1, , v_I)$, capa oculta $\mathbf{h} = (h_1, , h_J)$, distribución conjunta $p(\mathbf{v}, \mathbf{h}) = \exp(-E(\mathbf{v}, \mathbf{h}))$ y función de energía $E(\mathbf{v}, \mathbf{h}) = -\mathbf{h}^T \boldsymbol{W} \mathbf{v} - \mathbf{c}^T \mathbf{v} - \boldsymbol{b}^T \mathbf{h}$. Demuestre que la distribución marginal sobre la capa visible se puede escribir de la siguiente forma:

$$p(\mathbf{v}) = Z^{-1} \exp\left(\mathbf{c}^T \mathbf{x} + \sum_{j=1}^{J} s\left(\boldsymbol{W}_{j\cdot} \mathbf{v} + b_j \right) \right) ,$$

donde Z es la función de partición de la RBM, $\boldsymbol{W}_{j\cdot}$ denota la j-ésima fila de la matriz $\boldsymbol{W}$ y $s(\cdot)$ es la función *softplus*, es decir, $s(\xi) = \ln(1 + \exp(\xi))$. Derive desde acá la expresión para la log-verosimilitud del modelo generativo.

14. ⋆⋆ (25 puntos) Considere un un autoencoder ordinario (AE) donde el "encoder" se implementa como $h(\mathbf{x}) = s(\boldsymbol{W}\mathbf{x} + \boldsymbol{b})$ y el "decoder" se implementa como $g(\mathbf{z}) = s(\boldsymbol{W'}\mathbf{z} + \boldsymbol{b'})$; $\boldsymbol{W} \in \mathbb{R}^{J \times I}, \boldsymbol{W'} \in \mathbb{R}^{I \times J}$. Demuestre que si utiliza una función de activación lineal $s(\xi) = \xi$ y se mide el error de reconstrucción usando la función de pérdida cuadrática $J(\boldsymbol{W}, \boldsymbol{W'}) = (\mathbf{x} - g(h(\mathbf{x})))^T (\mathbf{x} - g(h(\mathbf{x})))$, el encoder óptimo correspondiente a un cierto decoder fijo, se obtiene usando *tied weights*, es decir, $\boldsymbol{W'} = \boldsymbol{W}^T$.

15. (20 puntos) Considere una pequeña red neuronal recurrente con input $x_t \in \mathbb{R}$, capa oculta $z_t = \alpha_1 x_t + \beta_1 z_{t-1} + b_1$ y salida $y_t = \alpha_2 z_t + b_2$. Suponga que después del entrenamiento, los pesos de la red son $\alpha_1 = \beta_1 = \alpha_2 = 1$ y $b_1 = b_2 = 0$. Haga un diagrama de la red indicando claramente los ciclos y pesos correspondientes a estas ecuaciones. Genere además una representación completamente "desenrollada" (unfolded) de la red en el tiempo para la secuencia $x_1 = 2$, $x_2 = -0.5$, $x_3 = 1$, $x_4 = 1$. ¿Qué hace esta pequeña red?

16. (20 puntos) Considere una red neuronal recurrente de 1 capa oculta con recurrencias desde la salida entrenada para producir una secuencia a partir de otra secuencia (many-to-many). Una técnica usada con frecuencia en este caso, consiste en entrenar la red usando *teacher forcing*, es decir, usando la salida correcta en cada tiempo como input para el tiempo sucesivo, en vez de la predicción de la red.

    (a) Demuestre que esto es óptimo si se adopta el criterio de estimación de máxima verosimilitud.

    (b) Explique porqué podría ser inconveniente entrenar a la red usando las salidas correctas en vez de sus propias predicciones y proponga un criterio para atenuar este efecto.

17. (10 puntos) Suponga que desea entrenar una red LSTM usando *Dropout* sobre los pesos recurrentes. Para una determinada secuencia, ¿Debe ser la máscara correspondiente compartida/fija en el tiempo?