Universidad Técnica Federico Santa María
Departamento de Informática

INF-477 Redes Neuronales Artificiales
Pauta Control 3

1 Instrucciones

- Este certamen debe ser resuelto individualmente, sin apuntes, en un máximo de 25 minutos. Su nota se obtendrá como la suma de los puntos obtenidos, siendo el puntaje máximo de cada pregunta indicado al inicio de cada enunciado.
- Entregue las respuestas a cada pregunta utilizando un lápiz de tinta indeleble, con letra clara y legible. Recuerde también escribir su nombre y rol en cada hoja.
- Escriba explícitamente cualquier supuesto que crea importante y todos los pasos intermedios que sean necesarios para llegar a un resultado.

2 Preguntas

1. (50 pts.) 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) Explique por qué este criterio es óptimo si se adopta el criterio de estimación de máxima verosimilitud.

   Respuesta: Considerando la secuencia de entrenamiento $\mathbf{x}_{1:T} = \mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T$ y su correspondiente secuencia de salida $y_{1:T} = y_1, y_2, \ldots, y_T$. Si Θ denota los parámetros del modelo, tenemos que:

$$\begin{aligned} P(y|\mathbf{x}, \Theta) = P(y_{1:T}|\mathbf{x}, \Theta) &= P(y_T|\mathbf{x}, y_{1:T-1}, \Theta) \cdot P(y_{1:T-1}|\mathbf{x}, \Theta) \\ &= P(y_T|\mathbf{x}, y_{1:T-1}, \Theta) \cdot P(y_{T-1}|\mathbf{x}, y_{1:T-2}, \Theta) \cdot P(y_{1:T-2}|\mathbf{x}, \Theta) \\ &= \ldots \\ &= \prod_{t=1}^{T} P(y_t|\mathbf{x}, y_1, \ldots, y_{t-1}, \Theta). \end{aligned}$$

   Por lo tanto, la función de log-verosimilitud de $y_{1:T}$ condicional a $x_{1:T}$ tendrá la forma

$$\ell(\Theta) = \sum_{t=1}^{T} logP(y_t|\mathbf{x}, y_1, \ldots, y_{t-1}, \Theta),$$

   lo que muestra que la función de pérdida a utilizar para un "time step" t, durante el entrenamiento de la red, debe considerar las verdaderas salidas $y_1, \ldots, y_{t-1}$ hasta el tiempo anterior. Por ejemplo, si $y_t \in \mathbb{R}$ y asumimos que $y_t \sim N(f(x_{1:t}, y_{1:t-1}, \Theta), I)$ con $f(x_{1:t}, y_{1:t-1}, \Theta)$ una red con recurrencias desde la salida, tenemos que la función de log-verosimilitud toma la forma,

$$\ell(\Theta) = \sum_{t=1}^{T} -(y_t - f(x_{1:t}, y_{1:t-1}, \Theta))^2,$$

que es equivalente a entrenar la red para minimizar la función de pérdida

$$Q(y_{1:t}, f(\mathbf{x}_{1:t}, y_{1:t-1}, \Theta)) = \sum_{t=1}^{T} (y_t - f(x_{1:t}, y_{1:t-1}, \Theta))^2.$$

Nótese que lo anterior corresponde a entrenar a la red con la pérdida cuadrática y teacher forcing activado. Tener el teacher forcing desactivado correspondería a entrenar la red para minimizar

$$Q(y_{1:t}, f(\mathbf{x}_{1:t}, y_{1:t-1}, \Theta)) = \sum_{t=1}^{T} (y_t - f(x_{1:t}, \hat{y}_{1:t-1}, \Theta))^2.$$

con $\hat{y}_t = f(\mathbf{x}_{1:t}, \hat{y}_{1:t-1})$, es decir, al momento de predecir la salida en el "time step" t, se consideran las predicciones de la red hasta ese momento $\hat{y}_{1:t-1}$ en vez de las verdaderas salidas $y_{1:t-1}$.

(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.
Respuesta: El inconveniente de esta estrategia es que en la práctica, la red no tiene las verdaderas salidas (test set) $y_{1:t-1}$ para predecir y_t y se alimentaría en cambio de $\hat{y}_{1:t-1}$. Una solución frecuente es entrenar ejecutando algunas epochs con teacher forcing activado y algunas con teacher forcing desactivado.

2. (50 pts.) Suponga que se implementa una red neuronal feed-forward para procesar secuencias de largo fijo $T = 1000$, con elementos $x_t \in \mathbb{R}^{10}$. Suponga que utiliza una única capa oculta de tamaño 100 y que la salida es un escalar. ¿Cuántos parámetros tiene este modelo con respecto a una red recurrente estándar (sin rezago) con la misma cantidad de neuronas ocultas diseñada para procesar la secuencia?

Respuesta: Para poder procesar la secuencia de entrada, la red feedforward debiese concatenar los $T = 1000$ "time steps" en un solo gran vector de entrada $\mathbf{x} \in \mathbb{R}^{Td}$. La capa oculta aplicaría una transformación de la forma $h = \sigma(U\mathbf{x} + b)$ con $U \in \mathbb{R}^{k \times Td}$ y $b \in \mathbb{R}^k$. La capa de salida procesaría h para producir la salida, que corresponde a $T = 1000$ y's, uno por cada "time step", aplicando una transformación de la forma $y = \sigma(V\mathbf{x} + c)$ con $V \in \mathbb{R}^{o \times k}$ y $c \in \mathbb{R}$. Por lo tanto, esta red tiene $k \times Td + k + oT \times k + oT = 100 \times 1000 \times 10 + 100 + 1 \times 1000 \times 100 + 1 \times 1000 = 1101100$ parámetros entrenables.
La red recurrente comparte las matrices $U \in \mathbb{R}^{k \times d}, W \in \mathbb{R}^{k \times k}, V \in \mathbb{R}^{o \times k}, b \in \mathbb{R}^k, c \in \mathbb{R}$ en todos los "time steps" $t = 1, 2, \ldots, T$. Por lo tanto, tiene $k \times d + k \times k + o \times k + k + 1 = 100 \times 10 + 100 \times 100 + 1 \times 100 + 100 + 1 = 11201$ parámetros entrenables.

3. (25 pts.) ¿Es cierto o es falso que el encoder es siempre una función determinista de x?

Respuesta: Falso. Por ejemplo, en un VAE el encoder es estocástico.