Universidad Técnica Federico Santa María
Departamento de Informática

INF-477 Redes Neuronales Artificiales
Control 1

1. (25 pts.) Explique la ventaja más importante de utilizar funciones de activación *ReLu* $s(\xi) = \max(0, \xi)$ en el diseño de redes neuronales. ¿Por qué el uso de esta función de activación suele verse como un modo de dotar a la red de representaciones latentes (ocultas) de largo variable?

   Porque se puede ver la red como un árbol de clasificación donde se inhiben o excitan ciertas variables latentes dependiendo del vector de entrada.

2. (25 pts.) Explique en qué consiste la técnica denominada *progressive decay* para el entrenamiento de redes neuronales artificiales.

   Consiste en ir bajando progresivamente por una tasa de decaimiento eta_d(en cada iteración) el valor de la tasa de aprendizaje (*learning rate*), comenzando en un valor inicial η_0. En la iteración s la tasa de aprendizaje se calcula como:

   $$\eta^(s) = \eta_0/(1 + s\eta_d)$$

3. (50 pts.) La función de pérdida denominada *cross-entropy* es una de las elecciones más comunes en el entrenamiento de redes neuronales artificiales para clasificación. Demuestre que usar esta función resulta equivalente a estimar los pesos de la red usando el método de *máxima verosimilitud*, paradigma clásico en el diseño de modelos estadísticos. Explique claramente el modelo de probabilidad que debe implementar la capa de salida para obtener este resultado.

   Sean X e Y las variables aleatorias correspondientes a la entrada y la respuesta del sistema que nos interesa modelar. En problemas de clasificación $Y|X$ sigue una distribución categórica con recorrido $[K] = \{1, 2, \ldots, K\}$ y parámetros $p_1, p_2, \ldots, p_K$, donde $p_k = P(Y = k|X)$ y $\sum_k p_k = 1$. Parece natural que cada neurona de salida modele directamente la probabilidad condicional de cada clase, esto es $p_k = f_k(\mathbf{x}; \theta)$. Esto se consigue por ejemplo usando una red neuronal con capa de salida softmax con K neuronas de salida.

   Consideremos el conjunto de entrenamiento $\{\mathbf{x}_m, y_m\}_{m=1}^{M}$, donde $y_m \in \{\mathbf{e}_1, \ldots, \mathbf{e}_K\}$ y $\mathbf{e}_i$ corresponde al i-ésimo vector de la base canónica. La función de verosimilitud condicional está dada por:

   $$\mathcal{L}(\mathbf{x}; \theta) = P(Y|X; \theta) = \prod_{m=1}^{M} p(y_m|\mathbf{x}_m, \theta) = \prod_{m=1}^{M} \prod_{k=1}^{K} f_k(\mathbf{x}_m; \theta)^{y_m^k}$$

   Maximizar el logaritmo de la función de verosimilitud equivale a minimizar

   $$-\sum_{m=1}^{M} \sum_{k=1}^{K} y_m^k \ln f_k(\mathbf{x}_m; \theta).$$

   Es decir, minimizar

   $$-\sum_{m=1}^{M} \ell(y_m, f_k(\mathbf{x}_m; \theta),$$

   donde $\ell(y_m, f_k(\mathbf{x}_m; \theta) = \sum_{k=1}^{K} y_m^k \ln f_k(\mathbf{x}_m; \theta)$.

CVV LaTeX