uc3m | Universidad Carlos III de Madrid

OPENCOURSEWARE
REDES DE NEURONAS ARTIFICIALES
Ricardo Aler

Preguntas y Ejercicios para Evaluación: Tema 7

1. ¿Por qué es difícil entrenar con backpropagation a una red de neuronas con muchas capas ocultas?

   Respuesta:

   El algoritmo de backpropagation se basa en la actualización de los pesos de cada capa propagando el error desde la capa de salida hasta la capa de entrada. Sin embargo, a medida que nos alejamos de la capa de salida, la actualización de los pesos se vuelve cada vez más lenta, de tal manera que los pesos de las capas más cercanas a la entrada, prácticamente no se modifican y por tanto no hay aprendizaje. A este problema se le conoce como "vanishing gradient": el gradiente que se utiliza para actualizar los pesos disminuye de manera exponencial a medida que nos alejamos de la capa de salida.

2. ¿Por qué dropout permite combatir el sobreaprendizaje?

   Respuesta:

   Con dropout, en distintas iteraciones se eliminan grupos de neuronas distintas. De esta manera, una neurona en algunas iteraciones tendrá las salidas de otra neurona determinada, y en otras iteraciones no la tendrá. Así, se fuerza a la neurona a no sobre-adaptarse a la presencia de las salidas de las neuronas que tiene a la entrada.

3. ¿Por qué al pre-entrenamiento de los pesos para redes con múltiples capas se le denomina "no-supervisado"?

   Respuesta:

   Porque no se utiliza la variable de respuesta (o sea, la salida, en un problema de clasificación o de regresión) para inicializar los pesos de las capas ocultas.

4. ¿Las capas más cercanas a la salida, computan características más simples o más complejas que aquellas más cercanas a la entrada?

Respuesta:

Las capas más lejanas a la entrada computan características más complejas que las más cercanas a la entrada, puesto que se basan en transformaciones no lineales computadas por las capas anteriores. En las redes convolucionales, la primera capa oculta detecta características en la entrada mediante los filtros. La siguiente capa de convolución detecta características en el mapa de características detectadas por el primer filtro, y así sucesivamente.

5. Explicar tres razones por las que una red convolucional puede ser más apropiada para clasificar imágenes, que una red normal con el mismo número de capas:

   Respuesta:

   1. Tienen simetría translacional. Es decir, permiten detectar una característica (un segmento vertical, por ejemplo) en cualquier lugar de la imagen
   2. Detectan características locales. Los filtros tienen un campo receptivo mucho más pequeño que la imagen completa y permiten centrarse en características localizadas (por ejemplo, un pequeño segmento vertical). Dado que en reconocimiento de imágenes, muchas características son locales, se evita la detección de características no-locales (por ejemplo, un pixel activo en la esquina inferior izquierda de la imagen y un pixel activo en la parte superior derecha de la imagen), que en muchos casos pueden ser espúreas.
   3. El número de pesos de una red convolucional es mucho más pequeño que el número de pesos de una red estándar (fully connected). Esto limita las posibilidades de sobre-aprendizaje.

6. Supongamos que queremos clasificar imágenes en blanco y negro, de 32x32 píxeles cada una. Suponiendo que una red convolucional tiene las siguientes capas, calcular el número de parámetros (pesos) total de la red:
   1. Una primera capa C1 convolucional, con 6 filtros de 5x5 y sin non-zero padding.
   2. Una segunda capa S2 de sub-sampling, con 6 filtros max-pooling de 2x2 y stride=2.
   3. Una tercera capa C3 de convolución, con 16 filtros de 5x5 y sin zero padding.
   4. Una cuarta capa S4 de sub-sampling, con 16 filtros max-pooling de 2x2 y stride=2.
   5. Una red fully connected con una capa de 120 neuronas (FC1), otra de 84 (FC2), y 10 neuronas de salida.

      Respuesta:

      - C1: cada filtro de 5x5 tiene 25 pesos, mas el bias. Como hay 6 filtros, C1 tiene (5x5+1)*6 = 156 pesos. Dado que la capa C3 tiene las mismas características, también tendrá **156 pesos**.

- S2: las capas de max-pooling no tienen pesos, se limitan a calcular el máximo en una región de 2x2. Lo mismo ocurre con S4
- Fully connected FC1: para saber el número de pesos que conectan S4 con la primera capa de la red FC que tiene 120 neuronas (FC1), necesitamos saber el tamaño de los mapas de características de S4. Para ello, necesitamos saber también los tamaños de las capas de características de C1, S2 y C3.
  - Como la entrada es de 32x32 píxeles y el mapa de características de C1 se genera "deslizando" (convolución) un filtro de 5x5, y no hay zero-padding, el mapa de características C1 será de 28x28 píxeles (32-5+1=28).
  - Cada uno de los 6 mapas de características de C1 reduce su dimensionalidad a la mitad gracias a los filtros 2x2 de S2. S2 por tanto tendrá 6 mapas de 14x14.
  - Los 6 mapas de características de C3 serán de 10x10 (14-5+1=10).
  - C4 reducirá la dimensionalidad a la mitad y tendremos a la salida de C4 6 mapas de 5x5.
  - Por tanto, a la entrada de FC1, tendremos 6x5x5 = 150 entradas
  - Por cada una de las 120 neuronas de FC1 tendremos 150 pesos, mas el bias. Un total de 120*(150+1) = **18120 pesos**
  - Por cada una de las 84 neuronas de FC2 tendremos 120+1 pesos. Un total por tanto de 84*(120+1) = **10164 pesos**.
  - Por cada una de las 10 neuronas de salida tendremos 84+1 pesos. Un total de 10*(84+1) = **850 pesos**.

- En total **2*156+18120+10164+850 = 29446 pesos.**

7. En el caso de la red anterior, ¿cuál sería el número de pesos si la red fuera fully connected, y las capas convolucionales fueran capas ocultas fully connected?

   Respuesta:

   En ese caso, según los resultados del ejercicio anterior, la primera capa oculta tendría 6*28*28 neuronas (primera capa convolucional del ejercicio anterior), la segunda capa oculta (segunda capa convolucional), tendría 6*10*10 neuronas ocultas. Así, la primera capa oculta tendría (32*32+1)*(6*28*28) = **4821600 pesos**. La segunda capa oculta tendría (6*28*28+1)*(6*10*10) = **2823000 pesos**. En total, la red fully connected tendría **4821600+2823000+18120+10164+850=7673734 pesos.** Podemos ver que una red convolucional tiene muchos menos pesos (29446) que la red fully connected equivalente (7673734).