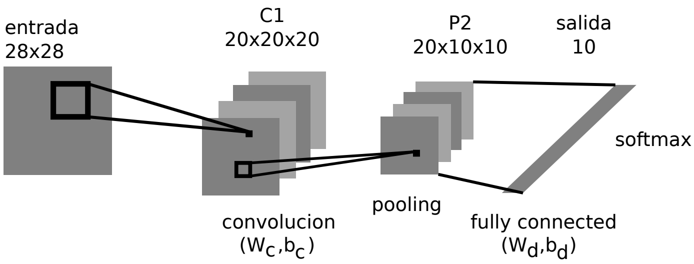
Métodos Avanzados de Síntesis y Procesamiento de Imágenes

3er Bimestre 2025

Laboratorio N° 3: Redes Convolucionales.

Vamos a implementar un aprendizaje de una red CNN como muestra la figura



Los datos de entrada de la red son imágenes de dígitos (LPR) en niveles de gris con un tamaño 28x28 pixeles.

- Descargar de link[1] el archivo **svmsmo_1.tar.gz** (source code).
- Descomprimir el archivo en un folder temporal.
- Recuperar el folder con las imágenes de los dígitos en[2] y copiarlo en el folder labo/datos.

Ejercicio 1. El script *Ej1.py* testea la implementación de la convolución en el archvo *convolve.py*. Este archivo se entrega en la Tarea del labo.

Ejercicio 2. *Ej2.py* nos permite testear el código que realiza el pooling de las matrices. Se debe completar el código en el archivo *pooling.py*. Este archivo se entrega en la tarea del laboratorio.

Ejercicio 3. En una segunda etapa del latoratorio, vamos a implementar el aprendizaje de la red convolucional completo a partir de estos datos.

Los parámetros son los pesos y bias de la capa convolucional (W_c, b_c) y los pesos y bias de la capa densa (W_d, b_d).

En la función cnnCost se calculan los gradientes de los parámetros de la red de la siguiente manera:

1. Propagar las imágenes de entrada guardando en probs las salidas. Usar eq. 1.

$$P(y^{(i)} = k|\mathbf{z}^{(i)}; W^{F6}, b^{F6}) = \frac{e^{W_k^{F6}\ \mathbf{z}+b_k^{F6}}}{\sum_j^K e^{W_j^{F6}\ \mathbf{z}+b_j^{F6}}} \tag{1}$$

[1] http://www.ipol.im/pub/art/2018/173/
[2] svm_smo/SVMCode/Datasets/BaseOCR_MultiStyle

2. Calcular el costo usando eq. 2 y guardarlo en cost.

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K}\exp(\theta^{(j)\top} x^{(i)})}\right] \tag{2}$$

3. Calcular el error de salida del softmax usando $E^{(i)}(\theta; x^{(i)}, y^{(i)})$ de eq. 3.

$$\begin{aligned}
\nabla_{\theta^{(k)}} J(\theta) &= -\sum_{i=1}^{m}\left[x^{(i)}\left(1\{y^{(i)} = k\} - P(y^{(i)} = k | x^{(i)}; \theta)\right)\right] \\
&= -\sum_{i=1}^{m}\left[x^{(i)} E^{(i)}(\theta; x^{(i)}, y^{(i)})\right]
\end{aligned} \tag{3}$$

4. $\Delta W_d = \frac{1}{m}\sum_i E^{(i)} * a^{(P2)}$, donde $a^{(P2)}$ es la activación de la capa de pooling.
5. $\Delta b_d = \frac{1}{m}\sum_i E^{(i)}$
6. Upsample de $\delta^{(P2)} = W_d \bullet E^{(i)}$.
7. Calcular el $\delta^{(C1)} = \delta^{(P2)} \bullet f'(z)$ (eq. 4).

$$\delta_k^{(l)} = \text{upsample}\left((W_k^{(l)})^T \delta_k^{(l+1)}\right) \bullet f'(z_k^{(l)}) \tag{4}$$

8. $\Delta W_c = \frac{1}{m}\sum_i x^{(i)} * \text{rot90}(\delta^{(C1)})$, siendo $x^{(i)}$ la imagen de entrada.
9. $\Delta b_c = \frac{1}{m}\sum \delta^{(C1)}$