

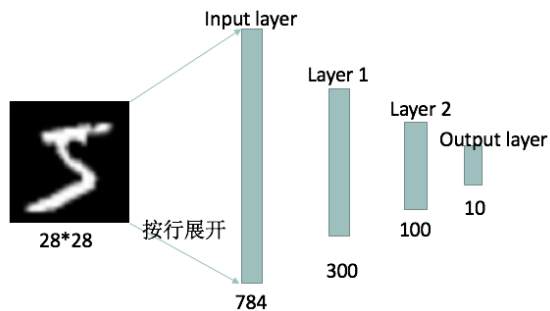
### 1. 数据读取：

- 此次实验的数据集是手写数字图片数据集 MNIST 的一个子集，其中训练集有 10000 张图片，测试集有 1000 张图片。
- 读取：使用 Python 的 cPickle 模块。

### 2. 数据预处理

原始图片每个像素点  $x \in [0, 255]$ ，在实验中为了更好的优化我们将  $x$  除以 256 (对训练集和测试集)。

### 3. 网络结构：3 层全连接神经网络



### 4. 网络选择：

- 激活函数：layer1 和 layer2 采用 Rectified Linear Unit (RELU)  $f(x) = \max(0, x)$  作激活函数，Output layer 采用 softmax

$$\sigma(\mathbf{z})_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad \text{for } j = 1, \dots, K.$$

作激活函数。

$$W \sim U\left[-\frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}, \frac{\sqrt{6}}{\sqrt{n_j + n_{j+1}}}\right]$$

- 权值初始化：Xavier 初始化，即权

值初始化自一个均匀分布， $n_j$  和  $n_{j+1}$  分别为权值矩阵的行数和列数（即上一层和此层神经元数目），详见 Glorot Xavier 的论文“Understanding the difficulty of training deep feedforward neural networks.”和

<https://zhuanlan.zhihu.com/p/22028079>.

- 损失函数：cross entropy loss function, 即  $E_n(\mathbf{w}) = -\sum_{i=1}^C y_i \ln a_i$ ，其中  $\mathbf{y}$  表示 label 的 1-of-C 形式，C 为类别数（也是输出层神经元数目）； $\mathbf{a}$  表示输出层的输出向量。

### 5. 训练算法：mini-batch 模式的梯度下降

- 在这种模式下，总的损失函数为  $L = \frac{1}{N} \sum_{n=1}^N E_n(\mathbf{w}) + \frac{\lambda}{2} \mathbf{w}^2$ ，其中 N 为 batchsize，即每训练 N 张图片才更新一次权值。 $\lambda$  为正则化参数。

- b) 本次实验中， $N=100$ ， $\lambda = 0.0005$ ， $\text{batchsize}=100$ ，迭代次数  $\text{iterations}=100$ ，即  $\text{batchsize} \times \text{iterations}=10000$ ，刚好遍历数据集，而遍历数据集的次数即  $\text{epochs}=100$
- c) 权值的更新公式为： $w \leftarrow w - \eta \frac{\partial L}{\partial w}$ ，其中  $\eta$  为学习率，本次实验中  $\eta$  的初始值为 0.1，当  $\text{epochs} \geq 50$  时， $\eta$  的值取初始值的  $\frac{1}{10}$

## 6. 输出要求：

- a) 在每个 epoch 训练结束后记录一个 batch（即最后一个 batch）的 Loss  $\frac{1}{N} \sum_{n=1}^N E_n(w)$  的值，最后应该得到一个 100 维向量，用 plot 命令画出 Loss 相对于 epoch 的变化曲线。提交 Loss 向量和曲线图。
- b) 在每个 epoch 训练结束后在 test 集上测试并记录准确率，最后应该得到一个 100 维向量，用 plot 命令画出准确率相对于 epoch 的变化曲线。提交准确率向量和曲线图。
- c) python 画图：使用 matplotlib 包