Part1

Data: $X \in R^d, Y \in \{0, \dots, K-1\}$
Number of hidden layers: $M > 1$
Number of hidden units per layer: L
Parameter: $\theta = \{W^1, b^1, \dots, W^{M+1}, b^{M+1}\}$
Softmax function: g()
Keep probability: p
Learning rate: α

Training:
Select (x, y) at random from dataset

Forward
$h(z) = \big(\sigma(z_1), \dots, \sigma(z_L)\big)$
$r_j^m \sim iid.Bernulli(p)$

$$Z_1 = W^1 X + b^1$$
$$a^2 = h(Z_1)$$
$$\tilde{a}^2 = (r_1^2 a_1^2, \dots, r_L^2 a_L^2)$$

$$Z_2 = W^2 a^2 + b^2$$
$$\dots$$
$$Z_M = W^M \tilde{a}^M + b^M$$
$$a^{M+1} = h(Z_M)$$
$$\tilde{a}^{M+1} = (r_1^{M+1} a_1^{M+1}, \dots, r_L^{M+1} a_L^{M+1})$$

$$Z_{M+1} = W^{M+1} \tilde{a}^{M+1} + b^{M+1}$$
$$f_\theta(x, y) = \big(g(Z_{M+1})\big)$$

Backward

$$logL(\theta) = \sum_{i=1}^N log f_\theta(x^i, y^i), l(\theta, x, y) = log f_\theta(x, y)$$
$$\delta_i^{M+2} = \frac{\partial l}{\partial {Z_i}^{M+1}}(\theta, x, y) = 1_{\{y=i\}} - f_\theta(x, i)$$
$$\delta_i^m = \frac{\partial l}{\partial {Z_i}^{m-1}}(\theta, x, y) \qquad 1 < m \le M+1$$

when 1 < m < M + 1,

$$\delta_i^m = \sum_{j=1}^L \frac{\partial l}{\partial {Z_i}^{M+1}}(\theta, x, y) \frac{\partial {Z_j}^m}{\partial {Z_i}^{m-1}} = \sum_{j=1}^L \delta_j^{m+1} W_{ji}^m r_i^m \sigma'(Z_i^{m-1})$$

when m = M + 1,

$$\delta_i^{M+1} = \sum_{j=1}^{K} \delta_j^{M+2} W_{ji}^{M+1} r_i^{M+1} \sigma'(Z_i^M)$$

$$\frac{\partial l}{\partial {W_{ij}}^m}(\theta, x, y) = \delta_i^{m+1} \tilde{\alpha}_j^m \qquad 1 \le m \le M+1$$

$$\frac{\partial l}{\partial {b_i}^m}(\theta, x, y) = \delta_i^{m+1} \qquad m \in (1, M+1)$$

$$\theta_{n+1} = \theta_n + \alpha \frac{\partial l}{\partial \theta}(\theta, x, y)$$

Testing:

$$Z_1 = W^1 X + b^1$$
$$\alpha^2 = h(Z_1)$$

$$Z_2 = pW^2 \alpha^2 + b^2$$
$$\dots$$
$$Z_M = pW^M \alpha^M + b^M$$
$$\alpha^{M+1} = h(Z_M)$$

$$Z_{M+1} = pW^{M+1} \alpha^{M+1} + b^{M+1}$$
$$f_\theta(x, y) = \big(g(Z_{M+1})\big)$$

Part2

2 convolution layers with pooling
I submitted the original .txt and .py files. Here just copy the results:

test accuracy 0.9924 with keep probability of 0.5
test accuracy 0.9897 without dropout

Part3

For convenience, I just submitted the 8_0.4_aug.py (layers: 8, keep prob: 0.4, with augmentation), 8_0.4_no.py (layers: 8, keep prob: 0.4, without augmentation) and 13_0.4_no.py (layers: 13, keep prob: 0.4, without augmentation) on campus 2g. Other files just have difference on keep probability. The accuracies are summarized in the following table.

Augmentation	Hidden layers	Keep prob.	Test accuracy
N	8 most layers have 300 units	0.4	0.801693
		0.6	0.801895
		0.8	0.797963
Y		0.4	0.864764
N	13 64 units	0.4	0.766043
		0.6	0.764701
		0.8	0.751312

Even if the number of hidden layers is 8, smaller than 13, the associated models have more accurate results because of much more hidden units for each layer. The augmentation can improve the accuracy.