Midterm Formula Sheet

CS540

July 5, 2019

1 Formulas

- Perceptron activation function: $a_i = \mathbb{1}_{\{w^T x_i + b \geqslant 0\}}$
- Perceptron update rule: $w = w - \alpha \left(a_i - y_i\right) x_i$ and $b = b - \alpha \left(a_i - y_i\right)$
- Logistic regression cost function: - $\sum_{i=1}^{n} \left(y_i \log\left(f\left(x_i\right)\right) + \left(1 - y_i\right) \log\left(1 - f\left(x_i\right)\right)\right)$
- Logistic regression activation function: $a_i = \dfrac{1}{1 + \exp\left(-w^T x_i - b\right)}$
- Logistic regression update rule: $w = w - \alpha \sum_{i=1}^{n} \left(a_i - y_i\right) x_i$ and $b = b - \alpha \sum_{i=1}^{n} \left(a_i - y_i\right)$
- Eigenvalues: $Av = \lambda v$
- Gradients for 2 layer neural network with logistic activation and squared error cost:

$\dfrac{\partial C}{\partial w_{j'j}^{(1)}} = \sum_{i=1}^{n} \left(a_i - y_i\right) a_i \left(1 - a_i\right) w_j^{(2)} a_{ij}^{(1)} \left(1 - a_{ij}^{(1)}\right) x_{ij}$' and $\dfrac{\partial C}{\partial b_{j'}^{(1)}} = \sum_{i=1}^{n} \left(a_i - y_i\right) a_i \left(1 - a_i\right) w_j^{(2)} a_{ij}^{(1)} \left(1 - a_{ij}^{(1)}\right)$ and $\dfrac{\partial C}{\partial w_j^{(2)}} = \sum_{i=1}^{n} \left(a_i - y_i\right) a_i \left(1 - a_i\right) a_{ij}^{(1)}$ and $\dfrac{\partial C}{\partial b^{(2)}} = \sum_{i=1}^{n} \left(a_i - y_i\right) a_i \left(1 - a_i\right)$

- Gradients for general neural network with logistic activation and squared error cost: given $\delta_i^{(L)} = \left(a_i - y_i\right) a_i \left(1 - a_i\right)$ and $\delta_{ij}^{(l)} = \sum_{j'=1}^{m^{(l+1)}} \delta_{j'}^{(l+1)} w_{jj'}^{(l+1)} a_{ij}^{(l)} \left(1 - a_{ij}^{(l)}\right), l = 1, 2, ..., L-1$ update $\dfrac{\partial C}{\partial w_{j'j}^{(l)}} = \sum_{i=1}^{n} \delta_{ij}^{(l)} a_{ij'}^{(l-1)}, l = 1, 2, ..., L$ and $\dfrac{\partial C}{\partial b_j^{(l)}} = \sum_{i=1}^{n} \delta_{ij}^{(l)}, l = 1, 2, ..., L$
- Neural network update rule: for $l = 1, 2, ..., L$, update $w_{j'j}^{(l)} \leftarrow w_{j'j}^{(l)} - \alpha \dfrac{\partial C}{\partial w_{j'j}^{(l)}}, j' = 1, 2, ..., m^{(l-1)}, j = 1, 2, ..., m^{(l)}$ and $b_j^{(l)} \leftarrow b_j^{(l)} - \alpha \dfrac{\partial C}{\partial b_j^{(l)}}, j = 1, 2, ..., m^{(l)}$
- L_1 regularized cost for neural network: $\sum_{i=1}^{n} \left(a_i - y_i\right)^2 + \lambda \left(\sum_{i=1}^{m} \left|w_i\right| + |b|\right)$
- L_2 regularized cost for neural network: $\sum_{i=1}^{n} \left(a_i - y_i\right)^2 + \lambda \left(\sum_{i=1}^{m} w_i^2 + b^2\right)$

- SVM cost function: $\min_{w} \frac{\lambda}{2} w^T w + \frac{1}{n} \sum_{i=1}^{n} \max \left\{0, 1 - (2y_i - 1)\left(w^T x_i + b\right)\right\}$

- Gradient for soft margin SVM:

$$\partial_w C \ni \lambda w - \sum_{i=1}^{n} (2y_i - 1) x_i \mathbb{1}_{\{(2y_i-1)(w^T x_i + b) \geqslant 1\}} \text{ and } \partial_b C \ni - \sum_{i=1}^{n} (2y_i - 1)) \mathbb{1}_{\{(2y_i-1)(w^T x_i + b) \geqslant 1\}}$$

- Subgradient: $\partial f(x) = \left\{v : f(x') \geqslant f(x) + v^T (x' - x) \ \forall\, x'\right\}$

- SVM kernel matrix: $K_{ii'} = \phi(x_i)^T \phi(x_{i'})$

- Entropy formula: $H(Y) = - \sum_{y=1}^{K} p_y \log_2(p_y)$

- Conditional entropy formula: $H(Y|X = x) = - \sum_{y=1}^{K_Y} p_{y|x} \log_2\left(p_{y|x}\right)$ and $H(Y|X) = \sum_{x=1}^{K_X} p_x H(Y|X = x)$

- Information gain: $I(Y|X) = H(Y) - H(Y|X)$

- L_p norm: $\rho(x, x') = \left(\sum_{j=1}^{m} \left|x_j - x'_j\right|^p\right)^{\frac{1}{p}}$

- One dimensional convolution: $a = (a_1, a_2, ..., a_m) = x * w$ with $a_j = \sum_{t=-k}^{k} w_t x_{j-t}, j = 1, 2, ..., m$

- Two dimensional convolution: $A = X * W$ with $A_{j,j'} = \sum_{t=-k}^{k} \sum_{t'=-k}^{k} W_{t,t'} X_{j-t,j'-t'}, j, j' = 1, 2, ..., m$

- Image gradient: magnitude $G = \sqrt{\nabla_x^2 + \nabla_y^2}$ and direction $\Theta = \arctan\left(\frac{\nabla_y}{\nabla_x}\right)$

- N gram assumption: $\mathbb{P}\{z_t | z_{t-1}, z_{t-2}, ...\} = \mathbb{P}\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\}$

- N gram MLE estimate: without smoothing $\hat{\mathbb{P}}\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\} = \frac{c_{z_{t-N+1}, z_{t-N+2}, ..., z_t}}{c_{z_{t-N+1}, z_{t-N+2}, ..., z_{t-1}}}$ and with Laplace smoothing $\hat{\mathbb{P}}\{z_t | z_{t-1}, z_{t-2}, ..., z_{t-N+1}\} = \frac{c_{z_{t-N+1}, z_{t-N+2}, ..., z_t} + 1}{c_{z_{t-N+1}, z_{t-N+2}, ..., z_{t-1}} + m}$

- Marginal distribution: $\mathbb{P}\{X_j = x_j\} = \sum_{x \in X_{j'}} \mathbb{P}\{X_j = x_j, X_{j'} = x\}$

- Conditional distribution: $\mathbb{P}\{X_j = x_j | X_{j'} = x_{j'}\} = \frac{\mathbb{P}\{X_j = x_j, X_{j'} = x_{j'}\}}{\mathbb{P}\{X_{j'} = x_{j'}\}}$

- Joint probability in a Bayesian network: $\mathbb{P}\{X_1 = x_1, X_2 = x_2, ..., X_m = x_m\} = \prod_{j=1}^{m} \mathbb{P}\{X_j = x_j | p(X_j) = p(x_j)\}$

- Conditional probability table MLE estimate: without smoothing $\hat{\mathbb{P}}\{x_j | p(X_j)\} = \frac{c_{x_j, p(X_j)}}{c_{p(X_j)}}$ and with Laplace smoothing $\hat{\mathbb{P}}\{x_j | p(X_j)\} = \frac{c_{x_j, p(X_j)} + 1}{c_{p(X_j)} + |X_j|}$