ASSIGNMENT3

- 机器学习&神经网络
  - Adam优化器
  - Dropout层
- 依存句法分析
  - 代码实现
  - 错误样例分析

(A题)ADAM

- 相较SGD？动量方法（i题）+自适应步长（ii题）；

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta})$$

$$\mathbf{m} \leftarrow \beta_1 \mathbf{m} + (1-\beta_1)\nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta})$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \mathbf{m}$$

$$\mathbf{m} \leftarrow \beta_1 \mathbf{m} + (1-\beta_1)\nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta})$$
$$\mathbf{v} \leftarrow \beta_2 \mathbf{v} + (1-\beta_2)(\nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \odot \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}))$$
$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \odot \mathbf{m}/\sqrt{\mathbf{v}}$$

动量方法

- (i) Briefly explain (you don't need to prove mathematically, just give an intuition) how using m stops the updates from varying as much and why this low variance may be helpful to learning, overall.

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \longrightarrow \begin{aligned} \mathbf{m} &\leftarrow \beta_1 \mathbf{m} + (1-\beta_1) \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \\ \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} - \alpha \mathbf{m} \end{aligned}$$

① m通过指数衰减平均维护历史梯度，使更新方向保持一种趋势，从而梯度变化更稳定：
在山谷处收到干扰小，轨迹更稳；在鞍点处有机会冲出局部最优解；

② 低方差对应到参数更新波动小，学习更加稳定，即：
1) 使网络能更优和更稳定的收敛；
2) 减少振荡过程。

自适应步长

- (ii) Since Adam divides the update by $\sqrt{\text{v}}$, which of the model parameters will get larger updates? Why might this help with learning?

$$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} - \alpha \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \longrightarrow \begin{aligned} \mathbf{m} &\leftarrow \beta_1 \mathbf{m} + (1-\beta_1)\nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \\ \mathbf{v} &\leftarrow \beta_2 \mathbf{v} + (1-\beta_2)(\nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta}) \odot \nabla_{\boldsymbol{\theta}} J_{\text{minibatch}}(\boldsymbol{\theta})) \\ \boldsymbol{\theta} &\leftarrow \boldsymbol{\theta} - \alpha \odot \mathbf{m}/\sqrt{\mathbf{v}} \end{aligned}$$

① 梯度平方的指数衰减平均越小的参数会获得更大的更新步长；

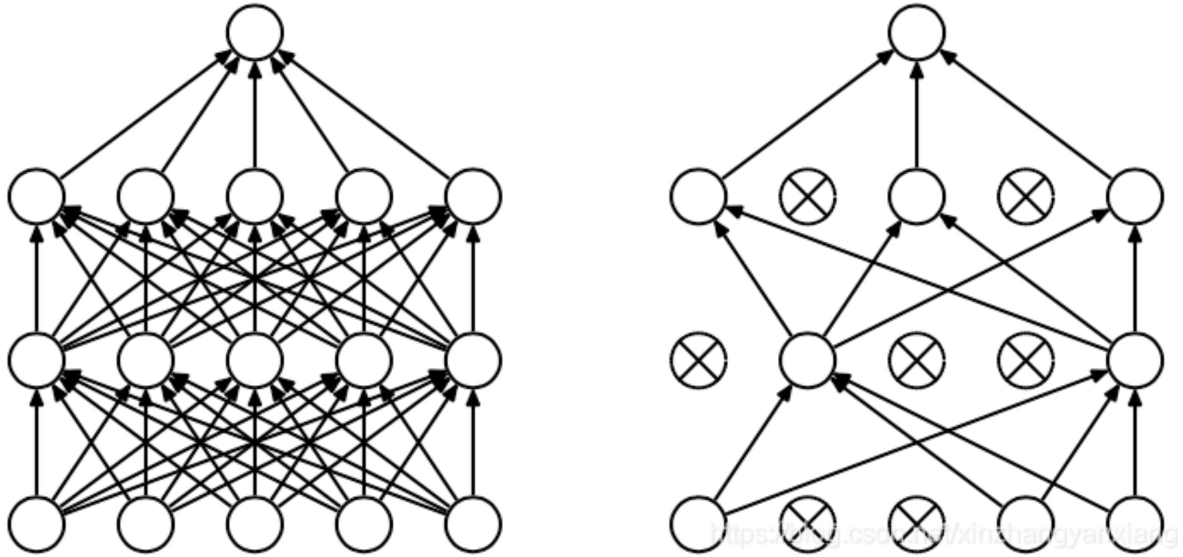
② 对于梯度较小的参数，应用较大的步长增加学习速度；对于梯度较大的参数，应用较小的步长保持学习的稳定性；使得不同梯度的参数保持在统一的更新量上，在稳定性和效率上做出平衡；

（B题） DROPOUT

- Dropout是一种正则化技术；

$$\mathbf{h}_{\text{drop}} = \gamma \mathbf{d} \circ \mathbf{h}$$

$$\mathbb{E}_{p_{\text{drop}}}[\mathbf{h}_{drop}]_i = h_i$$



DROPOUT层

- (i) What must γ equal in terms of p_drop? Briefly justify your answer.

$$\mathbf{h}_{\text{drop}} = \gamma \mathbf{d} \circ \mathbf{h}$$

$$\mathbb{E}_{p_{\text{drop}}}[\mathbf{h}_{drop}]_i = h_i$$

$$E_{p_{drop}}\left[h_{drop}\right]_i = \gamma\left(p_{drop} * 0 * h_i + \left(1 - p_{drop}\right) * 1 * h_i\right) = \gamma\left(1 - p_{drop}\right)h_i = h_i$$

$$\gamma = \frac{1}{1 - p_{drop}}$$

DROPOUT层

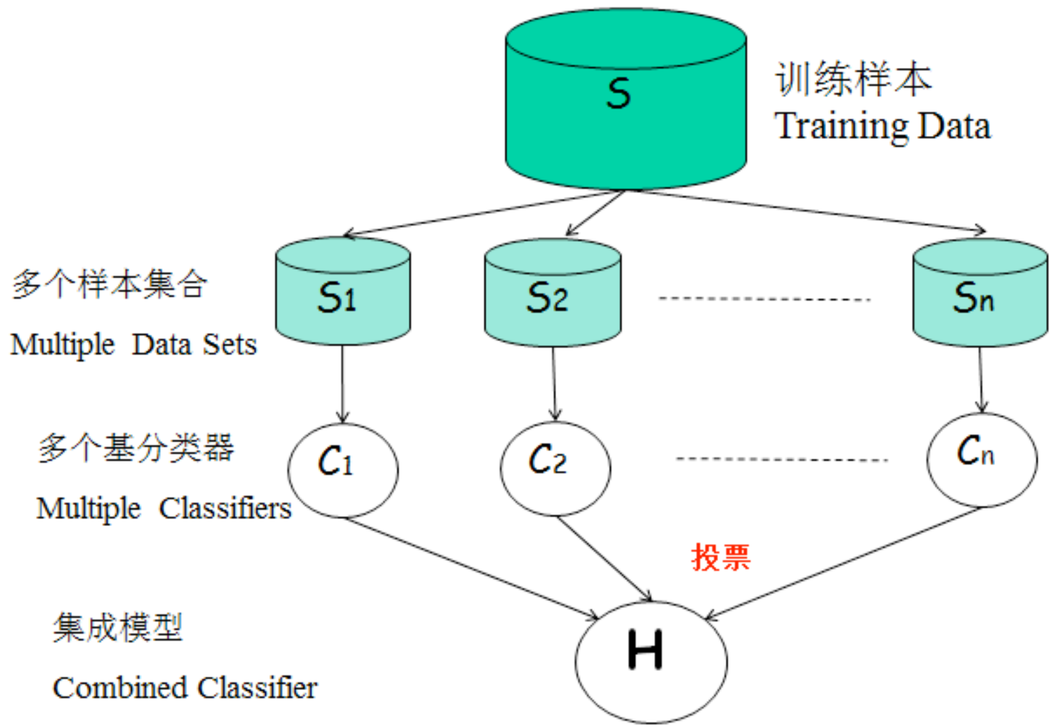
- (ii) Why should we apply dropout during training but not during evaluation?

Dropout层的动机：

(a) 达到一个取平均的作用；类似bagging；

(b) 减少神经元之间的共适应关系；
迫使网络更加鲁棒地去学习；

因此测试时不进行dropout，直接组装成一个大网络；
达到集成学习的效果；同时也避免模型的随机性；



DROPOUT层

- (ii) Why should we apply dropout during training but not during evaluation?

Dropout层的动机：

(a) 达到一个取平均的作用；类似bagging；

(b) 减少神经元之间的共适应关系；
迫使网络更加鲁棒地去学习；

因此测试时不进行dropout，直接组装成一个大网络；
达到集成学习的效果；同时也避免模型的随机性；