What does overfitting look like?

Later layers have more complex activations

- validation set MSE higher than train set MSE
- validation set accuracy lower than ~~validation~~ train set accuracy
- regression: predictions not smooth enough — fitting noise, not trend
- classification: decision boundary not smooth enough — fitting noise, not trend

---

~~Factors~~ Tools we have available to address overfitting:

- # layers, # units per layer
- Weight regularization
- Drop out

Process for finding a model:

1) Read the literature or find examples from a similar setting or application
2) Choose your best guess at a good starting point. Fit to training data & evaluate on validation data.
3) Increase model capacity until you are overfitting
4) Regularize model
   - Add L1 or L2 regularization
   - Add drop out
   - Remove layers/units
   - Reduce # of epochs (early stopping)

Can also tune things like learning rate, which optimizer you are using.

5) Refit to combined training & validation data and evaluate on test set.

L_2 regularization in terms of gradient calculations:

$$J(b,w) = \frac{1}{m}\sum_{i=1}^{m} J^{(i)}(b,w) + \sum_{l=1}^{L} \lambda^{[l]} \underbrace{\|w^{[l]}\|_2^2}_{\text{stands for } \sum_i \sum_j (w_{ij}^{[l]})^2}$$

$$\underbrace{\frac{\partial}{\partial w^{[l]}} J(b,w)}_{} = \left(\begin{array}{c}\text{stuff from}\\ \text{backpropagation}\end{array}\right) + \lambda^{[l]} \cdot 2 \cdot w^{[l]}$$

matrix of same shape as $w^{[l]}$ (n_{l-1}, n_l) where entry i,j is

$$\frac{\partial J(b,w)}{\partial w_{i,j}^{[l]}}$$

because

$$\frac{\partial}{\partial w_{i,j}^{[l]}} \sum_{l=1}^{L} \lambda^{[l]} \sum_i \sum_j (w_{ij}^{[l]})^2 = 2\lambda^{[l]} w_{ij}^{[l]}$$

So a gradient descent update step looks like:

$$w^{[l]} = w^{[l]} - \text{learning-rate} * \left(\begin{array}{c}\text{Stuff from}\\ \text{backpropagation}\end{array} + 2\lambda w^{[l]}\right)$$

$$= w^{[l]}\underbrace{(1 - 2\lambda * \text{learning-rate})}_{\substack{\text{every gradient descent}\\ \text{step tries to shrink}\\ \text{weights towards 0}\\ \text{(unless offset by a}\\ \text{gradient term)}}} - \text{learning-rate} * \left(\begin{array}{c}\text{stuff from}\\ \text{backpropagation}\end{array}\right)$$