Model complexity
Model complexity

• With polynomial regression, we saw an easy way to increase the complexity of our ML model.

  • With higher degree $d$, our model becomes strictly more powerful.

  • With larger coefficients, the regression line becomes more flexible.
Model complexity

- ML models can fail (i.e., exhibit poor accuracy) due to two reasons:

1. **Bias**: the model is too simple to fit the data distribution $\Rightarrow$ underfitting.

- This can result in low *training* accuracy.

![Graph showing underfitting](image-url)
Model complexity

- ML models can fail (i.e., exhibit poor accuracy) due to two reasons:

  2. **Variance**: the model is too complex and is prone to overfitting. Re-training on different datasets will result in very different weight values.

  - This can result in low *testing* accuracy.
Model complexity

• Note that the degree of polynomial regression is just one kind of model complexity.

• Others:
  
  • Size of the input (24x24? 36x36?) to the machine.
  
  • Number of layers in a neural network (more later).
Model complexity

- In general: the more training data you have...
  - ...the higher will be the testing accuracy of your trained machine.
  - ...the more complex of a model you can use without overfitting.
  - ...the less you need to regularize.
Model complexity

- In general: the more training data you have...
  - ...the higher will be the testing accuracy of your trained machine.
  - ...the more complex of a model you can use without overfitting.
  - ...the less you need to regularize.

- Therefore, as your training dataset grows, you might decide to switch to a more powerful architecture.
Illustration

• Simulation:
  • Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  • At each round, we add 4 more data points.
  • We compare polynomial regressors of degree 3 and 5.
Illustration

• Simulation:
  • Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  • At each round, we add 4 more data points.
  • We compare polynomial regressors of degree 3 and 5.
Illustration

- Simulation:
  - Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  - At each round, we add 4 more data points.
  - We compare polynomial regressors of degree 3 and 5.
Illustration

• Simulation:
  • Ground-truth: $y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x$
  • At each round, we add 4 more data points.
  • We compare polynomial regressors of degree 3 and 5.
Illustration

- Simulation:
  - Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  - At each round, we add 4 more data points.
  - We compare polynomial regressors of degree 3 and 5.
Illustration

- Simulation:
  - Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  - At each round, we add 4 more data points.
  - We compare polynomial regressors of degree 3 and 5.
Illustration

- Simulation:
  - Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  - At each round, we add 4 more data points.
  - We compare polynomial regressors of degree 3 and 5.
Illustration

- Simulation:
  - Ground-truth: \( y = -0.1x^5 + 0.1x^4 + 0.8x^3 - 1.8x^2 + 0.2x \)
  - At each round, we add 4 more data points.
  - We compare polynomial regressors of degree 3 and 5.

By this point, the poly5 regressor is better than the poly3 regressor.
Logistic regression
Using linear regression for classification

• In homework 2, you are using linear regression for classification.

  • **Regression**: predict any real number.

  • **Classification**: choose from a finite set (e.g., \( \{0, 1\} \)).

• While not incorrect, this is somewhat unnatural.
Using linear regression for classification

• During training, we penalize the linear regression model based on the MSE:

\[
\frac{1}{2n} \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2
\]

• Since every \( y \) is either 1 or 0, why let \( \hat{y} \) ever be greater than 1 or less than 0?

• Why not “squash” the output to always lie in (0,1)?
Sigmoid: a “squashing” function

• A sigmoid function is an “s”-shaped, monotonically increasing and bounded function.

• Here is the logistic sigmoid function $\sigma$:
Sigmoid: a “squashing” function

• A sigmoid function is an “s”-shaped, monotonically increasing and bounded function.

• Here is the **logistic sigmoid** function $\sigma$:

Which function(s) describe(s) the curve?

1. $\frac{e^x - e^{-x}}{e^x + e^{-x}}$
2. $\frac{e^x + e^{-x}}{e^x - e^{-x}}$
3. $\frac{1}{1 - e^{-x}}$
4. $\frac{1}{1 + e^{-x}}$
Sigmoid: a “squashing” function

- A sigmoid function is an “s”-shaped, monotonically increasing and bounded function.

- Here is the logistic sigmoid function $\sigma$:

\[
\sigma(x) = \frac{e^x}{1 + e^x}
\]

Which function(s) describe(s) the curve?

1. $\frac{e^x - e^{-x}}{e^x + e^{-x}}$  
   - tanh 
   - Similar but not quite right.

4. $\frac{1}{1 + e^{-x}}$
Sigmoid: a “squashing” function

- A sigmoid function is an “s”-shaped, monotonically increasing and bounded function.

- Here is the **logistic sigmoid** function $\sigma$:

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

Which function(s) describe(s) the curve?
Logistic sigmoid

• The logistic sigmoid function $\sigma$ has some nice properties:

• $\sigma(-z) = 1 - \sigma(z)$

\[
\begin{align*}
\sigma(z) &= \frac{1}{1 + e^{-z}} \\
1 - \sigma(z) &= 1 - \frac{1}{1 + e^{-z}} \\
&= \frac{1 + e^{-z}}{1 + e^{-z}} - \frac{1}{1 + e^{-z}} \\
&= \frac{e^{-z}}{1 + e^{-z}} \\
&= \frac{1}{1/e^{-z} + 1} \\
&= \frac{1}{1 + e^z} \\
&= \sigma(-z)
\end{align*}
\]
Logistic sigmoid

- The logistic sigmoid function $\sigma$ has some nice properties:

  - $\sigma'(z) = \sigma(z)(1 - \sigma(z))$

  \[
  \sigma(z) = \frac{1}{1 + e^{-z}} \\
  \frac{\partial \sigma}{\partial z} = \sigma'(z) = -\frac{1}{(1 + e^{-z})^2} (e^{-z} \times (-1)) \\
  = \frac{e^{-z}}{(1 + e^{-z})^2} \\
  = \frac{e^{-z}}{1 + e^{-z}} \times \frac{1}{1 + e^{-z}} \\
  = \frac{1}{1/e^{-z} + 1} \times \frac{1}{1 + e^{-z}} \\
  = \frac{1}{1 + e^z} \times \frac{1}{1 + e^{-z}} \\
  = \sigma(z)(1 - \sigma(z))
  \]
Logistic regression

- With logistic regression, our predictions are defined as:

  \[ \hat{y} = \sigma (\mathbf{x}^\top \mathbf{w}) \]

- Hence, they are forced to be in (0,1).

- For classification, we can interpret the real-valued outputs as probabilities that express how confident we are in a prediction, e.g.:

  - \( \hat{y} = 0.95 \): very confident that the class is a smile.
  - \( \hat{y} = 0.45 \): not very confident that the class is a non-smile.
Logistic regression

- How to train? Unlike linear regression, logistic regression has no analytical solution.
  - We can use gradient descent instead.
  - We have to apply the chain-rule of differentiation to handle the sigmoid function.
Gradient descent for logistic regression

- Let’s compute the gradient of $f_{\text{MSE}}$ for logistic regression.

- For simplicity, we’ll consider just a single example:

$$f_{\text{MSE}}(w) = \frac{1}{2}(\hat{y} - y)^2$$

$$= \frac{1}{2} \left( \sigma(x^\top w) - y \right)^2$$

$$\nabla_w f_{\text{MSE}}(w) = \nabla_w \left[ \frac{1}{2} \left( \sigma(x^\top w) - y \right)^2 \right]$$

$$= \left( \sigma(x^\top w) - y \right)$$
Gradient descent for logistic regression

• Let’s compute the gradient of $f_{\text{MSE}}$ for logistic regression.

• For simplicity, we’ll consider just a single example:

$$f_{\text{MSE}}(w) = \frac{1}{2} (\hat{y} - y)^2$$

$$= \frac{1}{2} (\sigma(x^\top w) - y)^2$$

$$\nabla_w f_{\text{MSE}}(w) = \nabla_w \left[ \frac{1}{2} (\sigma(x^\top w) - y)^2 \right]$$

$$= (\sigma(x^\top w) - y) \sigma(x^\top w) (1 - \sigma(x^\top w))$$
Gradient descent for logistic regression

- Let’s compute the gradient of $f_{\text{MSE}}$ for logistic regression.

- For simplicity, we’ll consider just a single example:

$$f_{\text{MSE}}(w) = \frac{1}{2} (\hat{y} - y)^2$$

$$= \frac{1}{2} (\sigma(x^T w) - y)^2$$

$$\nabla_w f_{\text{MSE}}(w) = \nabla_w \left[ \frac{1}{2} (\sigma(x^T w) - y)^2 \right]$$

$$= x (\sigma(x^T w) - y) \sigma(x^T w) (1 - \sigma(x^T w))$$
Gradient descent for logistic regression

- Let’s compute the gradient of $f_{\text{MSE}}$ for logistic regression.

- For simplicity, we’ll consider just a single example:

$$f_{\text{MSE}}(w) = \frac{1}{2} (\hat{y} - y)^2$$

$$= \frac{1}{2} (\sigma(x^T w) - y)^2$$

$$\nabla_w f_{\text{MSE}}(w) = \nabla_w \left[ \frac{1}{2} (\sigma(x^T w) - y)^2 \right]$$

$$= x (\sigma(x^T w) - y) \sigma(x^T w) \left(1 - \sigma(x^T w)\right)$$

$$= x (\hat{y} - y) \hat{y} (1 - \hat{y})$$
Gradient descent for logistic regression

- Let’s compute the gradient of \( f_{MSE} \) for logistic regression.

- For simplicity, we’ll consider just a single example:

\[
\begin{align*}
    f_{MSE}(w) &= \frac{1}{2}(\hat{y} - y)^2 \\
    &= \frac{1}{2} (\sigma(x^T w) - y)^2 \\
\end{align*}
\]

\[
\nabla_w f_{MSE}(w) = \nabla_w \left[ \frac{1}{2} (\sigma(x^T w) - y)^2 \right] \\
= x (\sigma(x^T w) - y) \sigma(x^T w) (1 - \sigma(x^T w)) \\
= x (\hat{y} - y) \hat{y} (1 - \hat{y})
\]

Notice the extra multiplicative terms compared to the gradient for linear regression: \( x(\hat{y} - y) \)
Attenuated gradient

- What if the weights $w$ are initially chosen badly, so that $\hat{y}$ is very close to 1, even though $y = 0$ (or vice-versa)?

  - Then $\hat{y}(1 - \hat{y})$ is close to 0.

- In this case, the gradient:

  $$\nabla_w f_{\text{MSE}}(w) = x (\hat{y} - y) \hat{y} (1 - \hat{y})$$

  will be very close to 0.

- If the gradient is 0, then no learning will occur!
Different cost function

• For this reason, logistic regression is typically trained using a different cost function from $f_{\text{MSE}}$.

• One particularly well-suited cost function uses logarithms.

• Logarithms and the logistic sigmoid interact well:

$$\frac{\partial}{\partial \mathbf{w}} \left[ \log \sigma (\mathbf{x}^\top \mathbf{w}) \right] =$$
Different cost function

• For this reason, logistic regression is typically trained using a different cost function from $f_{MSE}$.

• One particularly well-suited cost function uses logarithms.

• Logarithms and the logistic sigmoid interact well:

$$\frac{\partial}{\partial w} \left[ \log \sigma(x^\top w) \right] = \frac{1}{\sigma(x^\top w)} \sigma(x^\top w) \left(1 - \sigma(x^\top w)\right)$$
Different cost function

• For this reason, logistic regression is typically trained using a different cost function from $f_{MSE}$.

• One particularly well-suited cost function uses logarithms.

• Logarithms and the logistic sigmoid interact well:

\[
\frac{\partial}{\partial \mathbf{w}} \left[ \log \sigma(\mathbf{x}^\top \mathbf{w}) \right] = \frac{1}{\sigma(\mathbf{x}^\top \mathbf{w})} \sigma(\mathbf{x}^\top \mathbf{w}) (1 - \sigma(\mathbf{x}^\top \mathbf{w}))
\]

\[
= 1 - \sigma(\mathbf{x}^\top \mathbf{w})
\]

The gradient of log($\sigma$) is surprisingly simple.
Logarithm function

$\log(\hat{y})$ is undefined for $\hat{y} = 0$... but that's ok since $\hat{y} \in (0,1)$.
Log loss

• How could we define a “log-loss” function $f_{\log}$ so that:

  • $f_{\log}(y, \hat{y})$ is small when $\hat{y} \approx y$ and large when they are far apart.

1. $-y \log \hat{y} - \hat{y} \log y$

2. $-y \log \hat{y} - (1 - y) \log \hat{y}$

3. $-y \log \hat{y} - (1 - y) \log(1 - \hat{y})$

4. $-(1 - y) \log \hat{y} - y \log(1 - \hat{y})$
Log loss

- How could we define a “log-loss” function $f_{\log}$ so that:

- $f_{\log}(y, \hat{y})$ is small when $\hat{y} \approx y$ and large when they are far apart.

This expression is known as the log-loss.

3. $-y \log \hat{y} - (1 - y) \log(1 - \hat{y})$

The $y$ or $(1-y)$ “selects” which term in the expression is active, based on the ground-truth label.
Gradient descent for logistic regression with log-loss

\[ \nabla_{\mathbf{w}} f_{\log}(\mathbf{w}) = \nabla_{\mathbf{w}} \left[ - (y \log \hat{y} - (1 - y) \log(1 - \hat{y})) \right] \]
Gradient descent for logistic regression with log-loss

\[ \nabla_w f_{\text{log}}(w) = \nabla_w \left[ - (y \log \hat{y} - (1 - y) \log(1 - \hat{y})) \right] \\
= -\nabla_w \left( y \log \sigma(x^\top w) + (1 - y) \log(1 - \sigma(x^\top w)) \right) \]
Gradient descent for logistic regression with log-loss

\[
\nabla_w f_{\log}(w) = \nabla_w \left[ - (y \log \hat{y} - (1 - y) \log (1 - \hat{y})) \right] \\
= -\nabla_w \left( y \log \sigma(x^T w) + (1 - y) \log (1 - \sigma(x^T w)) \right) \\
= - \left( y \frac{x\sigma(x^T w)(1 - \sigma(x^T w))}{\sigma(x^T w)} - (1 - y) \frac{x\sigma(x^T w)(1 - \sigma(x^T w))}{1 - \sigma(x^T w)} \right)
\]
Gradient descent for logistic regression with log-loss

$$\nabla_w f_{\log}(w) = \nabla_w \left[ - (y \log \hat{y} - (1 - y) \log (1 - \hat{y})) \right]$$
$$= -\nabla_w \left( y \log \sigma(x^T w) + (1 - y) \log (1 - \sigma(x^T w)) \right)$$
$$= - \left( y \frac{x \sigma(x^T w)(1 - \sigma(x^T w))}{\sigma(x^T w)} - (1 - y) \frac{x \sigma(x^T w)(1 - \sigma(x^T w))}{1 - \sigma(x^T w)} \right)$$
$$= - (yx(1 - \sigma(x^T w)) - (1 - y)x \sigma(x^T w))$$
Gradient descent for logistic regression with log-loss

\[
\nabla_w f_{\text{log}}(w) = \nabla_w \left[ - (y \log \hat{y} - (1 - y) \log(1 - \hat{y})) \right] \\
= - \nabla_w \left( y \log \sigma(x^T w) + (1 - y) \log(1 - \sigma(x^T w)) \right) \\
= - \left( y \frac{x \sigma(x^T w)(1 - \sigma(x^T w))}{\sigma(x^T w)} - (1 - y) \frac{x \sigma(x^T w)(1 - \sigma(x^T w))}{1 - \sigma(x^T w)} \right) \\
= - \left( y x (1 - \sigma(x^T w)) - (1 - y) x \sigma(x^T w) \right) \\
= -x \left( y - y \sigma(x^T w) - \sigma(x^T w) + y \sigma(x^T w) \right) \\
= -x \left( y - \sigma(x^T w) \right) \\
= x(\hat{y} - y) \quad \text{Same as for linear regression!}
\]
## Linear regression versus logistic regression

<table>
<thead>
<tr>
<th></th>
<th>Linear regression</th>
<th>Logistic regression</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Primary use</strong></td>
<td>Regression</td>
<td>Classification</td>
</tr>
<tr>
<td><strong>Prediction (\hat{y})</strong></td>
<td>(\hat{y} = x^T w)</td>
<td>(\hat{y} = \sigma(x^T w))</td>
</tr>
<tr>
<td><strong>Loss</strong></td>
<td>(f_{MSE})</td>
<td>(f_{log})</td>
</tr>
<tr>
<td><strong>Gradient</strong></td>
<td>(x(\hat{y} - y))</td>
<td>(x(\hat{y} - y))</td>
</tr>
</tbody>
</table>
## Linear regression versus logistic regression

<table>
<thead>
<tr>
<th></th>
<th>Linear regression</th>
<th>Logistic regression</th>
</tr>
</thead>
<tbody>
<tr>
<td>Primary use</td>
<td>Regression</td>
<td>Classification</td>
</tr>
<tr>
<td>Prediction ($\hat{y}$)</td>
<td>$\hat{y} = x^Tw$</td>
<td>$\hat{y} = \sigma(x^Tw)$</td>
</tr>
<tr>
<td>Loss</td>
<td>$f_{MSE}$</td>
<td>$f_{log}$</td>
</tr>
<tr>
<td>Gradient</td>
<td>$x(\hat{y} - y)$</td>
<td>$x(\hat{y} - y)$</td>
</tr>
</tbody>
</table>

- Logistic regression is used primarily for classification even though it’s called “regression”.

- Logistic regression is an instance of a **generalized linear model** — a linear model combined with a **link function** (e.g., logistic sigmoid).

- In neural networks, link functions are typically called **activation functions**.
Multi-class ("polychotomous") classification
Multi-class classification

• So far we have talked about classifying only 2 classes (e.g., smile versus non-smile).

• This is sometimes called **binary classification**.

• But there are many settings in which multiple (>2) classes exist, e.g., emotion recognition, hand-written digit recognition:

  6 classes (fear, anger, sadness, happiness, disgust, surprise)  
  10 classes (0-9)
Classification versus regression

- Note that, even though the hand-written digit recognition (“MNIST”) problem has classes called “0”, “1”, …, “9”, there is no sense of “distance” between the classes.

- Misclassifying a 1 as a 2 is just as “bad” as misclassifying a 1 as a 9.
Multi-class classification

- It turns out that logistic regression can easily be extended to support an arbitrary number (>2) of classes.
  - The multi-class case is called **softmax regression**.
- How to represent the ground-truth $y$ and prediction $\hat{y}$?
  - Instead of just a scalar $y$, we will use a vector $y$. 
Example: 2 classes

• Suppose we have a dataset of 3 examples, where the ground-truth class labels are 0, 1, 0.

• Then we would define our ground-truth vectors as:

\[
\begin{align*}
y^{(1)} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix} \\
y^{(2)} &= \begin{bmatrix} 0 \\ 1 \end{bmatrix} \\
y^{(3)} &= \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\end{align*}
\]

• This is called one-hot encoding of the class label.