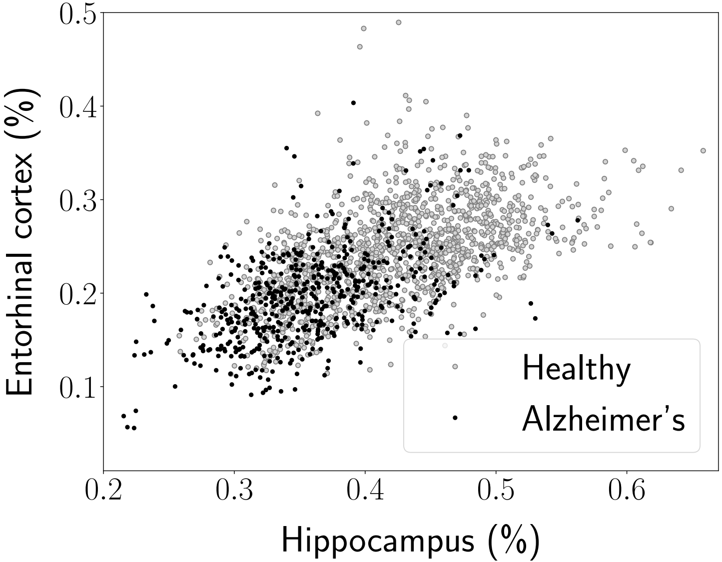
# Softmax Regression

Probability and Statistics for Data Science

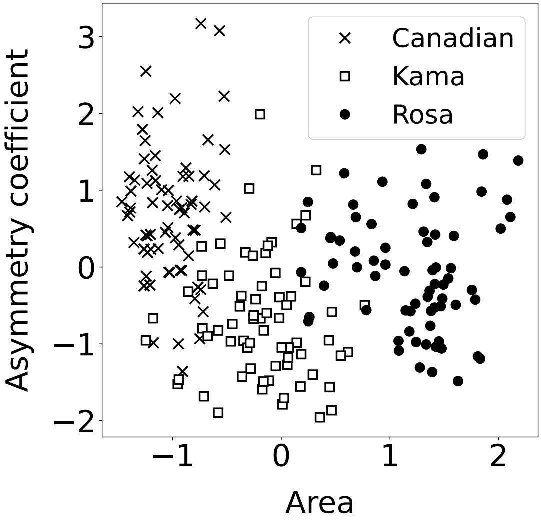
Carlos Fernandez-Granda

These slides are based on the book Probability and Statistics for Data Science by Carlos Fernandez-Granda, available for purchase here. A free preprint, videos, code, slides and solutions to exercises are available at https://www.ps4ds.net

# Binary classification



# Multiclass classification



# Classification

Data: $(x_1, y_1)$, $(x_2, y_2)$, ..., $(x_n, y_n)$

Each feature $x_i$ is a $d$-dimensional vector

The label $y_i$ indicates the class (e.g. *Canadian*, *Kama*, or *Rosa*)

Goal: Assign class to new data

# Probabilistic modeling

Model features as random vector $\tilde{x}$ and label as random variable $\tilde{y}$

For new data vector $x$:

$$\hat{y} := \arg \max_{y \in \{1,2,\ldots,c\}} p_{\tilde{y} \,|\, \tilde{x}}(y \,|\, x)$$

Is classification easy? No, due to curse of dimensionality!

# Discriminative classification

Goal: Use linear model to approximate $p_{\tilde{y}\,|\,\tilde{x}}(k\,|\,x)$ for $1 \leq k \leq c$ ($c \geq 2$ classes)

First idea:

$$\ell_k := \beta_k^T x + \alpha_k, \qquad 1 \leq k \leq c$$

Problem: For most values of $x$ *not a valid probability*

# Second idea

Use maximum

$$p_{\tilde{y}\,|\,\tilde{x}}(y\,|\,x) = \begin{cases} 1 & \text{if } \ell_y > \ell_k \qquad \text{for } 1 \leq k \leq c \\ 0 & \text{otherwise} \end{cases}$$

Problem 1: Does not encode uncertainty

Problem 2: Is not differentiable, impossible to estimate parameters!

# Softmax

Generalized linear model

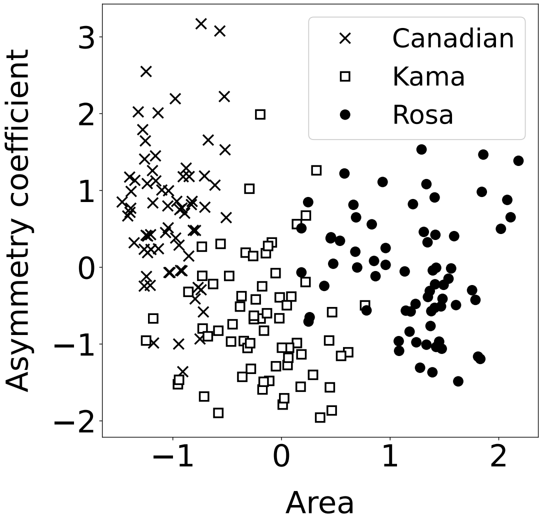
$$\mathrm{P}\left(\tilde{y} = k \,|\, \tilde{x} = x\right) = \frac{\exp\left(\ell_k\right)}{\sum_{l=1}^{c} \exp\left(\ell_l\right)}$$
$$= \frac{\exp\left(\beta_k^T x + \alpha_k\right)}{\sum_{l=1}^{c} \exp\left(\beta_l^T x + \alpha_l\right)} \qquad 1 \leq k \leq c$$

Differentiable

For $c := 2$ equivalent to logistic regression

Acts like a soft maximum

# Wheat varieties



$x_{\text{area}} := -1,\; x_{\text{asym}} := 2$

Logits:

$$\begin{bmatrix} \text{Canadian} \\ \text{Kama} \\ \text{Rosa} \end{bmatrix} = \begin{bmatrix} -7.7\, x_{\text{area}} + 0.9\, x_{\text{asym}} - 2.9 \\ 0.4\, x_{\text{area}} - 1.2\, x_{\text{asym}} + 2.7 \\ 7.3\, x_{\text{area}} + 0.4\, x_{\text{asym}} + 0.2 \end{bmatrix} = \begin{bmatrix} 6.6 \\ -0.1 \\ -6.3 \end{bmatrix}$$

$$\exp(6.6) = 735 \qquad \exp(-0.1) = 0.905 \qquad \exp(-6.3) = 0.002$$

After softmax:

$$\begin{bmatrix} \mathrm{P}(\text{Canadian} \mid x_{\text{area}}, x_{\text{asym}}) \\ \mathrm{P}(\text{Kama} \mid x_{\text{area}}, x_{\text{asym}}) \\ \mathrm{P}(\text{Rosa} \mid x_{\text{area}}, x_{\text{asym}}) \end{bmatrix} = \begin{bmatrix} \frac{735}{735+0.905+0.002} \\ \frac{0.905}{735+0.905+0.002} \\ \frac{0.002}{735+0.905+0.002} \end{bmatrix} = \begin{bmatrix} 0.999 \\ 0.001 \\ 0.000 \end{bmatrix}$$

# Parameter estimation

Data: $(x_1, y_1)$, $(x_2, y_2)$, ..., $(x_n, y_n)$

How do we estimate the softmax-regression parameters?

$$p_\Theta(x)_k := \frac{\exp\left(\beta_k{}^T x + \alpha_k\right)}{\sum_{l=1}^{c} \exp\left(\beta_l{}^T x + \alpha_l\right)}, \qquad 1 \leq k \leq c$$

Maximize the conditional likelihood of the labels given the features

# Likelihood

We model $i$th feature and label as random variables $\tilde{x}_i$ and $\tilde{y}_i$

Assumption 1:
Labels are conditionally independent given the features

Assumption 2:
$\tilde{y}_i$ is conditionally independent from $\{\tilde{x}_m\}_{m \neq i}$ given $\tilde{x}_i$

$$\begin{aligned}
\mathcal{L}_{XY}(\Theta) &:= \mathrm{P}\left(\tilde{y}_1 = y_1, \ldots, \tilde{y}_n = y_n \,|\, \tilde{x}_1 = x_1, \ldots, \tilde{x}_n = x_n\right) \\
&= \prod_{i=1}^{n} \mathrm{P}\left(\tilde{y}_i = y_i \,|\, \tilde{x}_1 = x_1, \ldots, \tilde{x}_n = x_n\right) \\
&= \prod_{i=1}^{n} \mathrm{P}\left(\tilde{y}_i = y_i \,|\, \tilde{x}_i = x_i\right) \\
&= \prod_{k=1}^{c} \prod_{\{i : y_i = k\}} p_\Theta\left(x_i\right)_k, \qquad \Theta := \{\beta_1, \ldots, \beta_c, \alpha_1, \ldots, \alpha_c\}
\end{aligned}$$

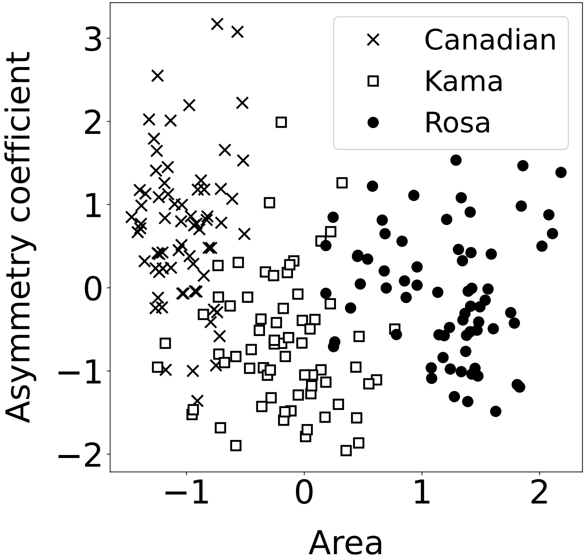
# Likelihood and log-likelihood

$$\mathcal{L}_{XY}(\Theta) = \prod_{k=1}^{c} \prod_{\{i : y_i = k\}} p_\Theta\left(x_i\right)_k$$

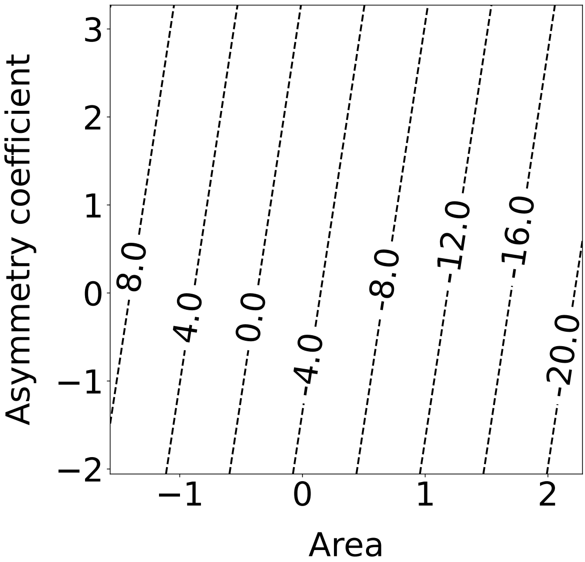
$$\log \mathcal{L}_{XY}(\Theta) = \sum_{k=1}^{c} \sum_{\{i : y_i = k\}} \log p_\Theta\left(x_i\right)_k$$

Maximized via iterative optimization methods, exploiting softmax differentiability

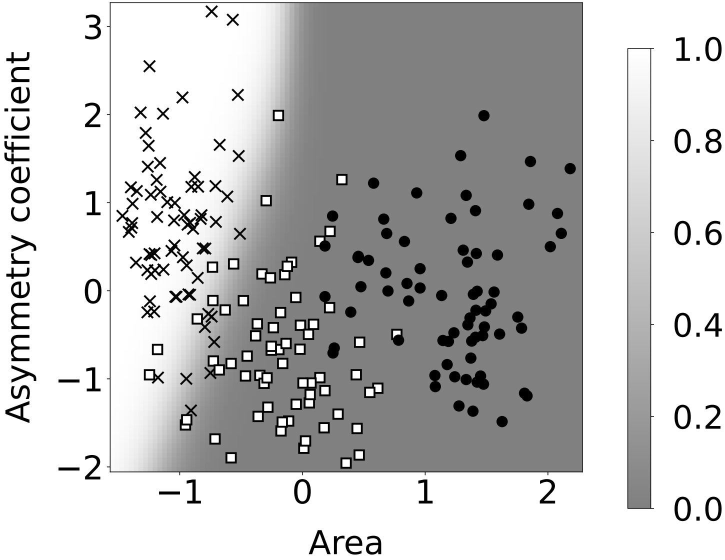
# Wheat varieties



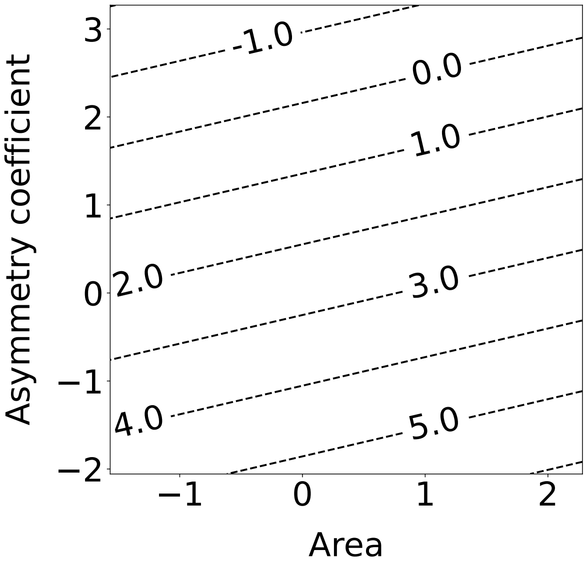
Canadian: $-7.7\,x_{\text{area}} + 0.9\,x_{\text{asym}} - 2.9$



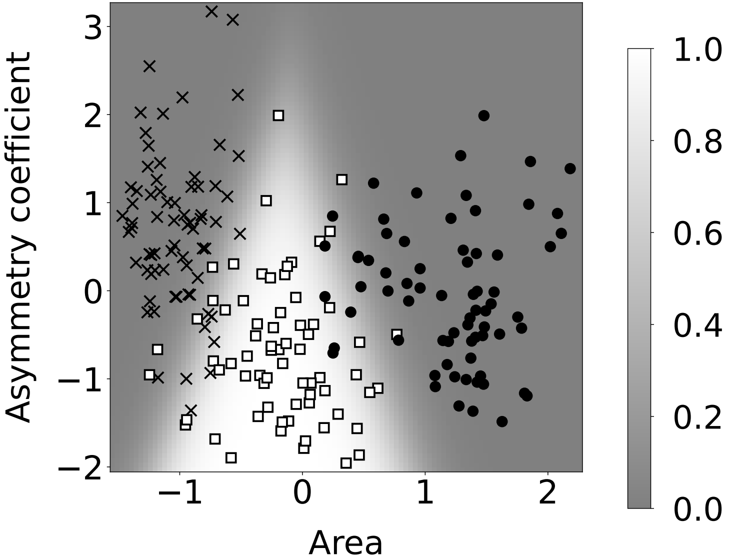
# Canadian: Estimated probability



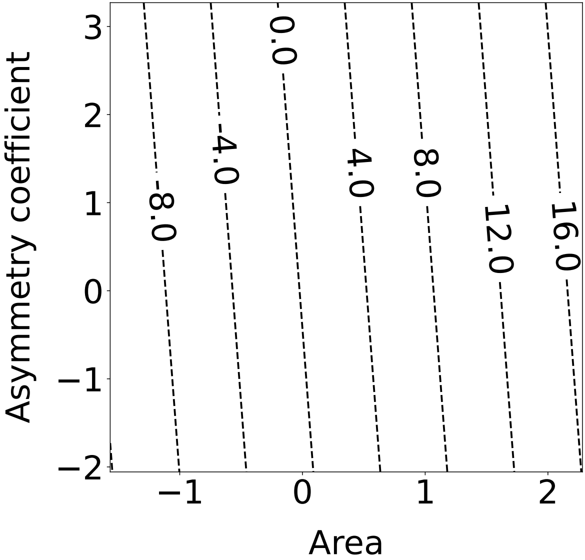
Kama: $0.4\,x_{\text{area}} - 1.2\,x_{\text{asym}} + 2.7$



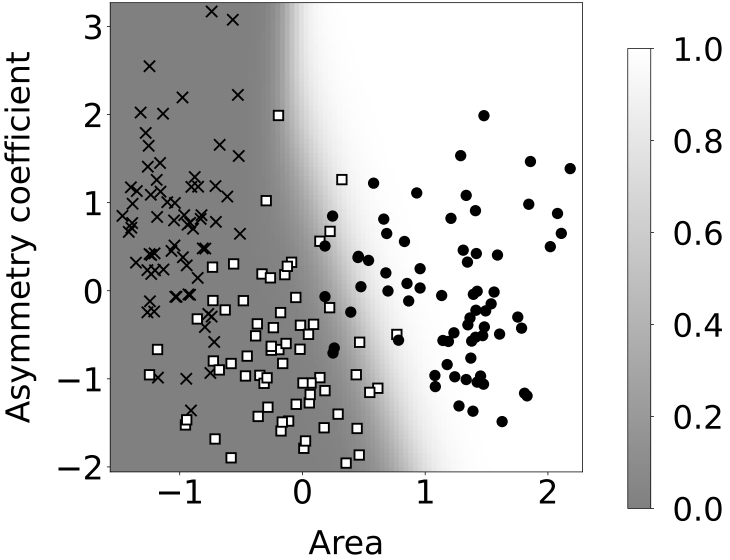
# Kama: Estimated probability



Rosa: $7.3\,x_{\text{area}} + 0.4\,x_{\text{asym}} + 0.2$



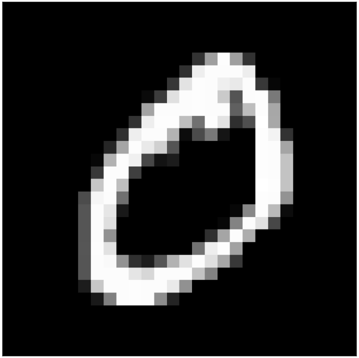
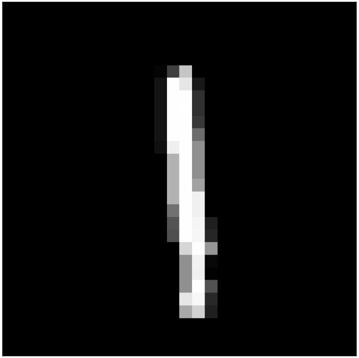
# Rosa: Estimated probability



# Digit classification

Goal: Classify 28 x 28 images of handwritten digits from the MNIST dataset

Training and test set: 35,000 examples each

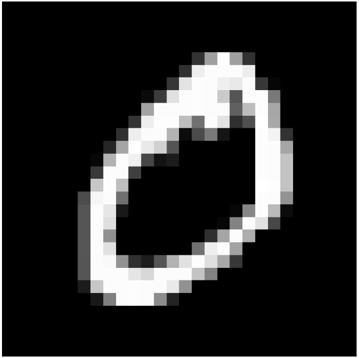
# Softmax regression

Training error: 4.3%

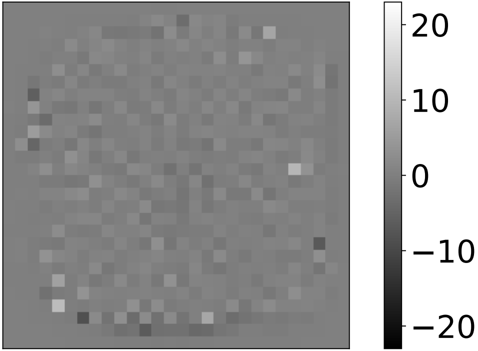
Test error: 10.4%

# Interpreting the model

$$\mathrm{P}\left(\tilde{y} = 0 \,|\, \tilde{x} = x\right) = \frac{\exp\left(\beta_0^T x + \alpha_0\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$



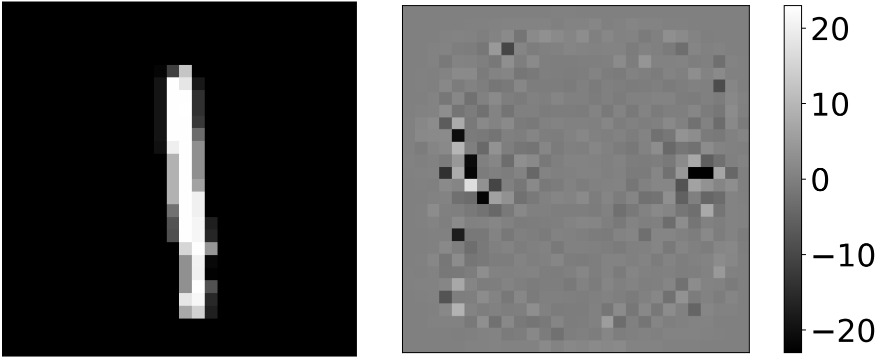
$x$




$\beta_0$

# Interpreting the model

$$\mathrm{P}\left(\tilde{y} = 1 \mid \tilde{x} = x\right) = \frac{\exp\left(\beta_1^T x + \alpha_1\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$




$\beta_1$

# What is going on?

Number of data = 35,000

Number of model parameters?

$\beta_k$, $0 \leq k \leq 9$: 10 (number of classes) $\times$ 784 (pixels) = 7840

$\alpha_k$, $0 \leq k \leq 9$: 10 (number of classes)

Overfitting!

Solution: Regularization

$$-\log \mathcal{L}_{XY}(\alpha, \beta) + \lambda \sum_{k=0}^{9} ||\beta_k||_2^2$$

where $\lambda$ is a regularization parameter

# Results

Without regularization:

Training error: 4.3%

Test error: 10.4%

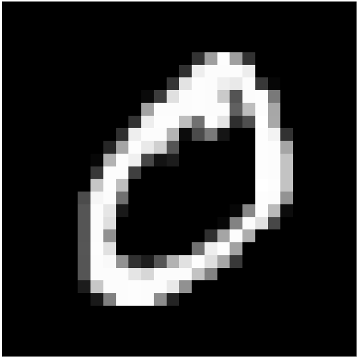
With regularization ($\lambda := 50$):

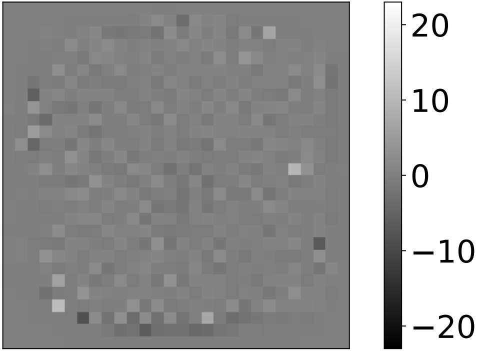
Training error: 6.2%

Test error: 7.8%

# Without regularization

$$\mathrm{P}\left(\tilde{y} = 0 \,|\, \tilde{x} = x\right) = \frac{\exp\left(\beta_0^T x + \alpha_0\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$

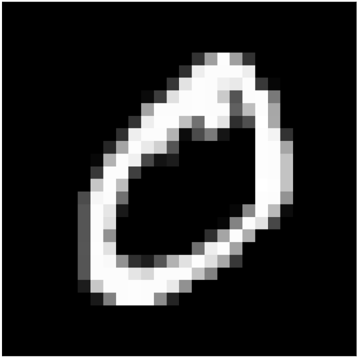


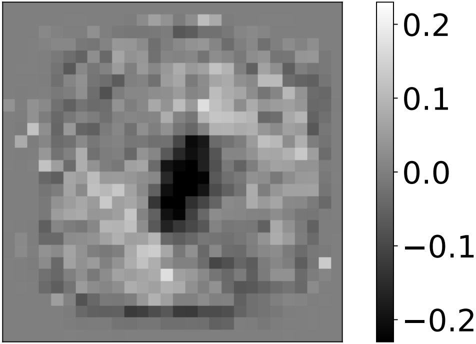



$\beta_0$

# With regularization

$$\mathrm{P}\left(\tilde{y} = 0 \,|\, \tilde{x} = x\right) = \frac{\exp\left(\beta_0^T x + \alpha_0\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$

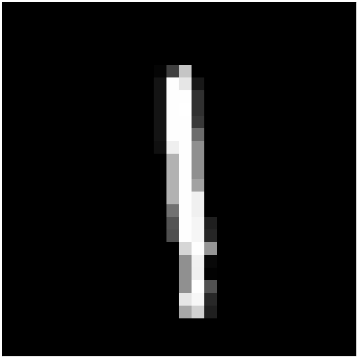


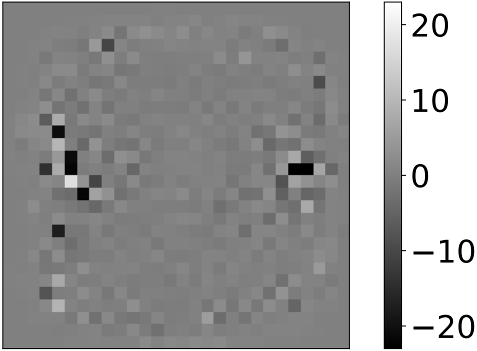



$\beta_0$

# Without regularization

$$\mathrm{P}\left(\tilde{y} = 1 \,|\, \tilde{x} = x\right) = \frac{\exp\left(\beta_1^T x + \alpha_1\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$

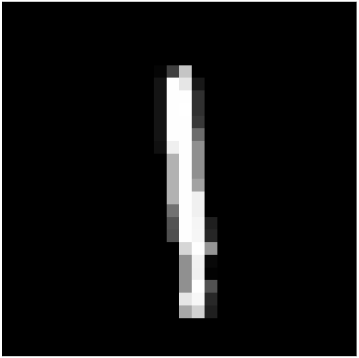


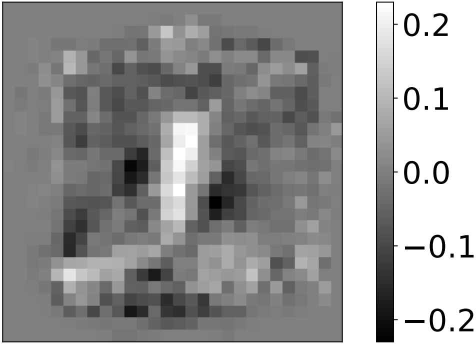



$\beta_1$

# With regularization

$$\mathrm{P}\left(\tilde{y} = 1 \,|\, \tilde{x} = x\right) = \frac{\exp\left(\beta_1^T x + \alpha_1\right)}{\sum_{l=0}^{9} \exp\left(\beta_l^T x + \alpha_l\right)}$$






$\beta_1$

# What have we learned?

How softmax regression works:

- Normalized exponential maps linear functions of features (logits) to probability estimates
- Parameters are obtained by maximizing the likelihood
- Regularization mitigates overfitting when data is scarce with respect to number of parameters