

# Introduction to machine learning and neural networks

Toby Dylan Hocking  
[toby.hocking@nau.edu](mailto:toby.hocking@nau.edu)  
[toby.hocking@r-project.org](mailto:toby.hocking@r-project.org)

July 6, 2020

## Introduction and applications

Practical considerations

Demonstration of overfitting in regression

Classifying images of digits

# Machine learning intro: image classification example

ML is all about learning predictive functions  $f(x) \approx y$ , where

- ▶ Inputs/features  $x$  can be easily computed using traditional algorithms, e.g. matrix of pixel intensities in an image.
- ▶ Outputs/labels  $y$  are what we want to predict, easy to get by asking a human, but hard to compute using traditional algorithms, e.g. image class.
- ▶ Input  $x$  = image of digit, output  $y \in \{0, 1, \dots, 9\}$ ,
  - this is a classification problem with 10 classes.



$$f(\text{0}) = 0$$



$$f(\text{1}) = 1$$

- ▶ Traditional/unsupervised algorithm: I give you a pixel intensity matrix  $x \in \mathbb{R}^{16 \times 16}$ , you code a function  $f$  that returns one of the 10 possible digits. Q: how to do that?

## Supervised machine learning algorithms

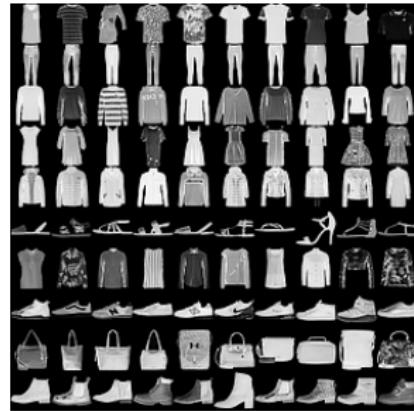
I give you a training data set with paired inputs/outputs, e.g.

0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9

Your job is to code an algorithm that learns the function  $f$  from the training data. (you don't code  $f$ )

Source: [github.com/cazala/mnist](https://github.com/cazala/mnist)

# Advantages of supervised machine learning

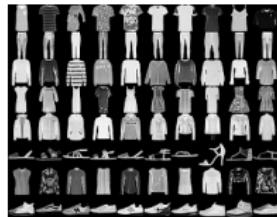


- ▶ Input  $x \in \mathbb{R}^{16 \times 16}$ , output  $y \in \{0, 1, \dots, 9\}$  types the same!
- ▶ Can use same learning algorithm regardless of pattern.
- ▶ Pattern encoded in the labels (not the algorithm).
- ▶ Useful if there are many un-labeled data, but few labeled data (or getting labels is long/costly).
- ▶ State-of-the-art accuracy (if there is enough training data).

# Learning two different functions

Say LEARN is a learning algorithm you have coded.

0	1	2	3	4	5	6	7	8	9
0	1	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9
0	1	2	3	4	5	6	7	8	9



$\text{LEARN}([0 \ 1 \ 2 \ 3 \ 4 \ 5 \ 6 \ 7 \ 8 \ 9]) \rightarrow f$ ,  $\text{LEARN}([\text{grid of images}]) \rightarrow g$



- ▶ Then we would expect  $f(\text{circle}) = 0$ ,  $f(\text{bar}) = 1$



- ▶  $g(\text{boot}) = \text{shoe}/0$ ,  $g(\text{pants}) = \text{pants}/1$



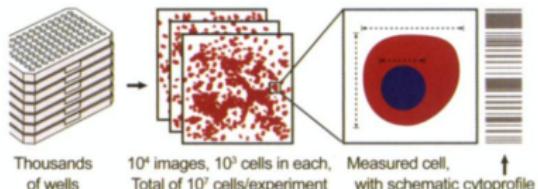
- ▶ Q: what happens if you do  $f(\text{boot})$ , or  $g(\text{circle})$ ?

# Machine learning for cell image classification (CellProfiler)

Jones *et al.* PNAS 2009. Scoring diverse cellular morphologies in image-based screens with iterative feedback and machine learning.

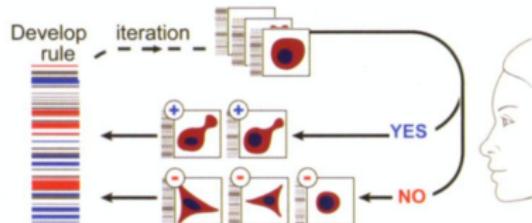
## A Automated Cell Image Processing

Cytoprofile of 500+ features measured for each cell



## B Iterative Machine Learning

System presents cells to biologist for scoring, in batches



- ▶ Input  $x$  = image of cell,
- ▶ Output  $y \in \{\text{yes, no}\}$  (binary classification),

- ▶  $f(\text{ } ) = \text{yes,}$
- ▶  $f(\text{ } ) = \text{no.}$

# Machine learning for image segmentation (LabelMe)

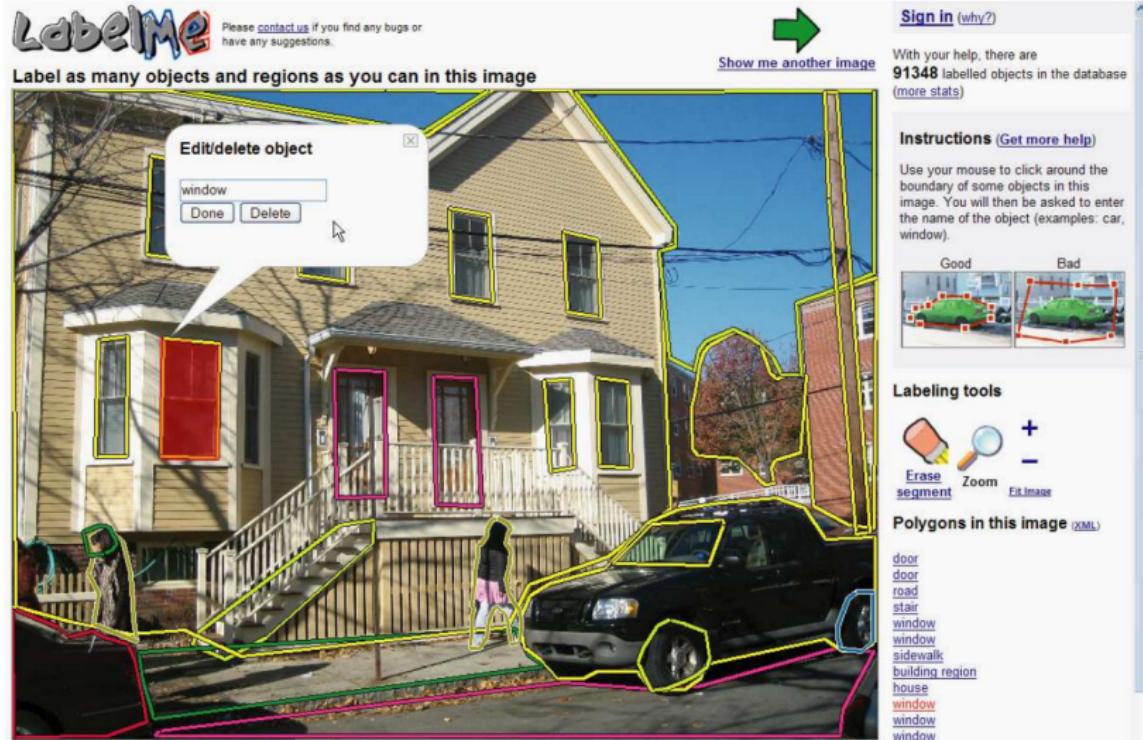


Figure 1. A screenshot of the labeling tool in use. The user is shown an image along with possibly one or more existing annotations, which are drawn on the image. The user has the option of annotating a new object by clicking along the boundary of the desired object and labeling it. Labels can be added to the image with the "Edit/delete object" dialog.



# Machine learning for image segmentation

Paper cup



Rock



Statue



Chair



Russell *et al.* 2007.

Q: What are the types/dimensions of  $x, y, f$  in this example?

# Machine learning for spam filtering (Gmail)

A few of your incoming messages has been suspended ➔ Spam

De Vito Raffael... Jan 11, 2019, 5:11 AM (3 days ago) ★ ↗  
to no\_reply@micorosoft.net

Why is this message in spam? It is similar to messages that were identified as spam in the past.

Report not spam

Re: Assay Standards call, September 5th

Myrto Kostadima ko... Tue, Sep 5, 2017, 1:45 AM ↗  
to Cath, assay\_standards@lhec-intranet.org

Please, find attached a short presentation on the ChIP-seq analysis pipeline.

Thanks,  
Myrto

Your Messages Has Been Suspended By Microsoft Outlook Team

A few of your incoming messages has been suspended by Microsoft verification Team because your email box account has not been properly verify. Do [verify](#) now to receive your suspended messages now.

On 01/09/2017 20:09, Cath Ennis wrote:

Hello all

We have a call scheduled for Tuesday September 5<sup>th</sup> at 6am Pacific time. I've attached the minutes from the last call (also available on the IHEC intranet)

Want:  $f(\text{email message}) \in \{0, 1\}$  – binary classification, spam=1 or not=0.

# Machine learning for translation (google books)

LE COMTE  
DE  
COUNT OF MONTE-CRISTO.  
MONTE-CRISTO  
BY  
ALEXANDRE DUMAS

ABRIDGED AND ANNOTATED  
BY  
EDGAR EWING BRANDON, A.M.  
*Professor of French in Miami University*

Twenty Illustrations,  
DRAWN ON WOOD BY M. VALENTIN,  
AND EXECUTED BY THE MOST EMINENT ENGLISH ENGRAVERS,  
UNDER THE SUPERINTENDENCE OF MR. CHARLES HEATH.



NEW YORK  
HENRY HOLT AND COMPANY  
1900

IN TWO VOLUMES.  
VOL. II.

LONDON:  
CHAPMAN AND HALL, 186 STRAND.

1846.

## Machine learning for translation (google translate)

— Fernand! lui cria-t-il, de mes noms, je n'aurais besoin de t'en dire qu'un seul pour te foudroyer; mais ce nom, tu le devines, n'est-ce pas? ou plutôt tu te le rappelles? car, malgré tous mes chagrins, toutes mes tortures, je te montre aujourd'hui un visage que le bonheur de la vengeance rajeunit, un visage que tu dois avoir vu bien souvent dans tes rêves depuis ton mariage... avec Mercédès, ma fiancée!

Le général, la tête renversée en arrière, les mains étendues, le regard fixe, dévora en silence ce terrible spectacle; puis, allant chercher la muraille comme point d'appui, il s'y glissa lentement jusqu'à la porte par laquelle il sortit à reculons, en laissant échapper ce seul cri lugubre, lamentable, déchirant:

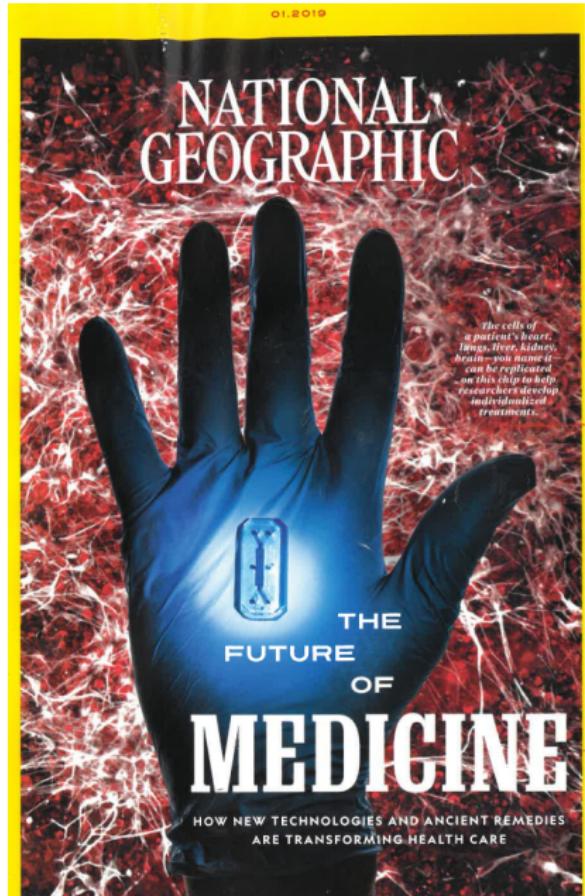
Want:  $f($  — Edmond Dantès!  $) =$

“Fernand,” cried he, “of my hundred names I need only tell you one to overwhelm you! But you guess it now; do you not?—or, rather, you remember it? For, notwithstanding all my sorrows and my tortures, I shew you to-day a face which the happiness of revenge makes young again—a face you must often have seen in your dreams since your marriage with Mercédès, my betrothed!”

The general, with his head thrown back, hands extended, gaze fixed, looked silently at this dreadful apparition; then seeking the wall to support him, he glided along close to it until he reached the door, through which he went out backwards, uttering this single mournful, lamentable, distressing cry,—

“Edmond Dantès!”

# Machine learning for medical diagnosis



f( )=

Thanks to artificial intelligence and machine learning, diagnostic tools can be trained to read tissue samples and radiologic scans. Google researchers fed more than a quarter-million patients' retinal scans into algorithms that recognize patterns—and the technology "learned" to spot which patterns predict a patient has high blood pressure or is at increased risk for heart attack or stroke. In some comparisons, digital tools produced more accurate analyses than did human pathologists, dermatolo-



Introduction and applications

Practical considerations

Demonstration of overfitting in regression

Classifying images of digits

## Machine learning algorithms in practice

To learn a good prediction function for any of these problems (and any other problems you encounter in your work) you should

- ▶ First gather a data set that you will use to train/test the machine learning algorithms, typically a CSV file with rows for observations and columns for features.
- ▶ Use K-fold Cross-Validation to repeatedly divide the observations into train/test sets. Use the train set to learn machine learning model parameters, and use the test set to estimate prediction error.
- ▶ Try a variety of different learning algorithms (e.g. linear models, random forests, boosting, neural networks, support vector machines, ...), because you don't know which one will result in most accurate predictions for any given data set.
- ▶ Each learning algorithm has different hyper-parameters which control complexity/regularization/overfitting, and typically must be chosen by the user (you) by dividing the train data into subtrain/validation sets. Use the hyper-parameter which is best with respect to the held-out validation set.

## K-fold cross-validation for evaluating prediction accuracy

Randomly create a vector of fold ID numbers from 1 to K, one fold ID for each observation:

```
set.seed(1) #for reproducibility.  
n.folds <- 4  
fold.vec <- sample(rep(1:n.folds, l=N.observations))
```

The result:

```
> str(fold.vec)  
int [1:100] 4 3 1 2 3 3 2 2 3 3 ...
```

## Hyper-parameter tuning

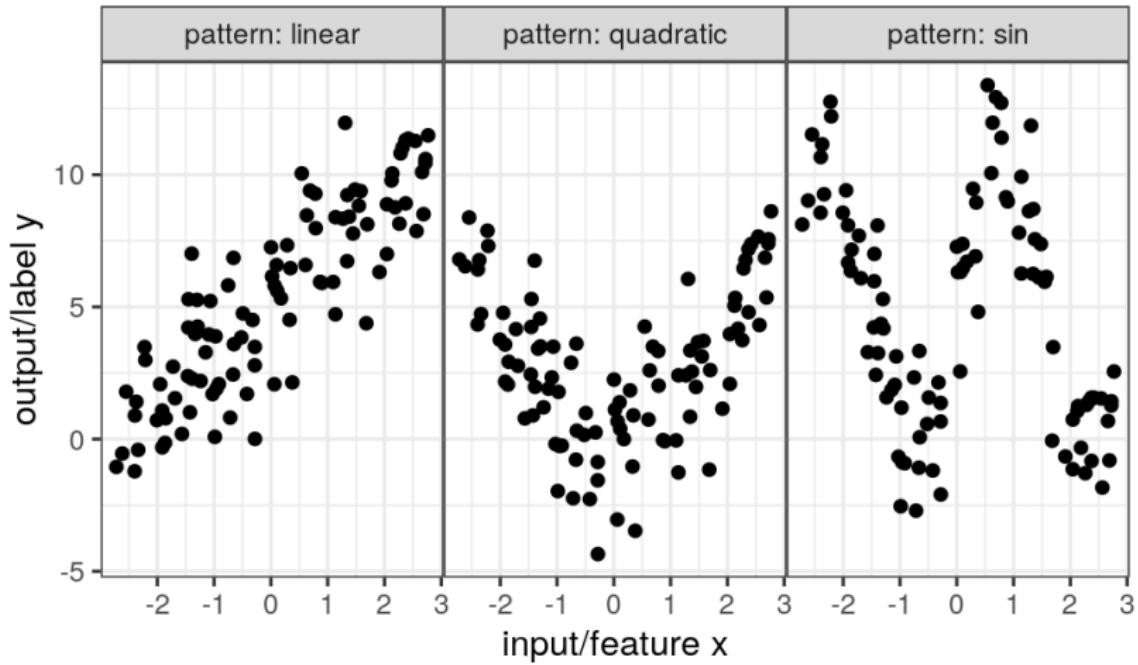
Fix a hyper-parameter value and use the subtrain data to fit use the subtrain data to learn a model for each of hyper-parameters, then use the hyper-parameters that result in maximal prediction accuracy with respect to the .

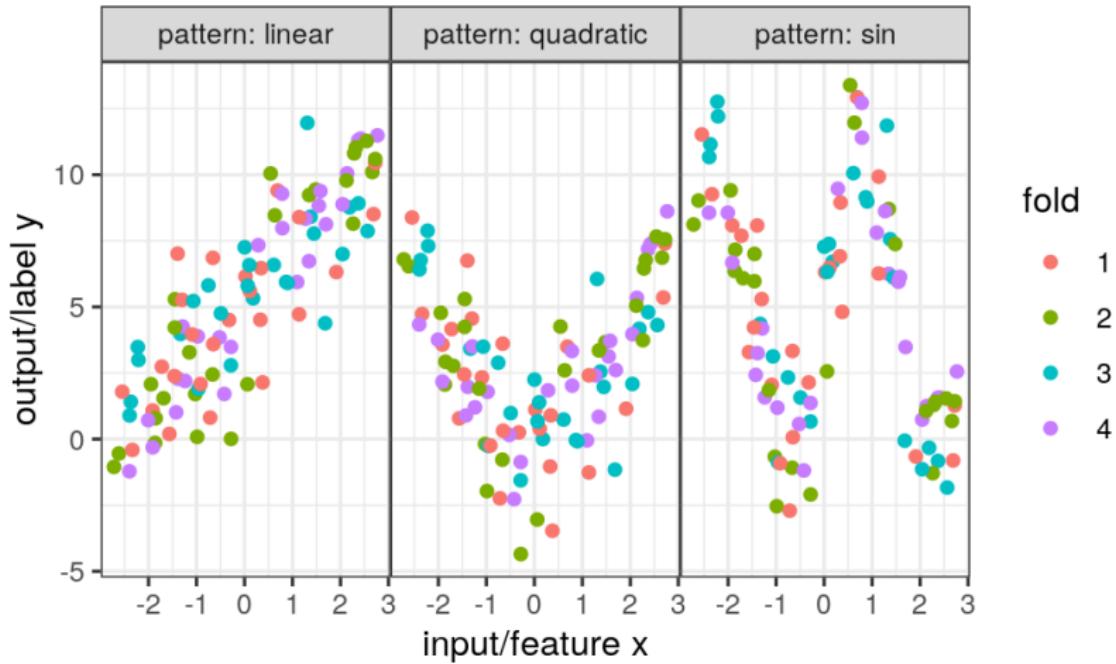
Introduction and applications

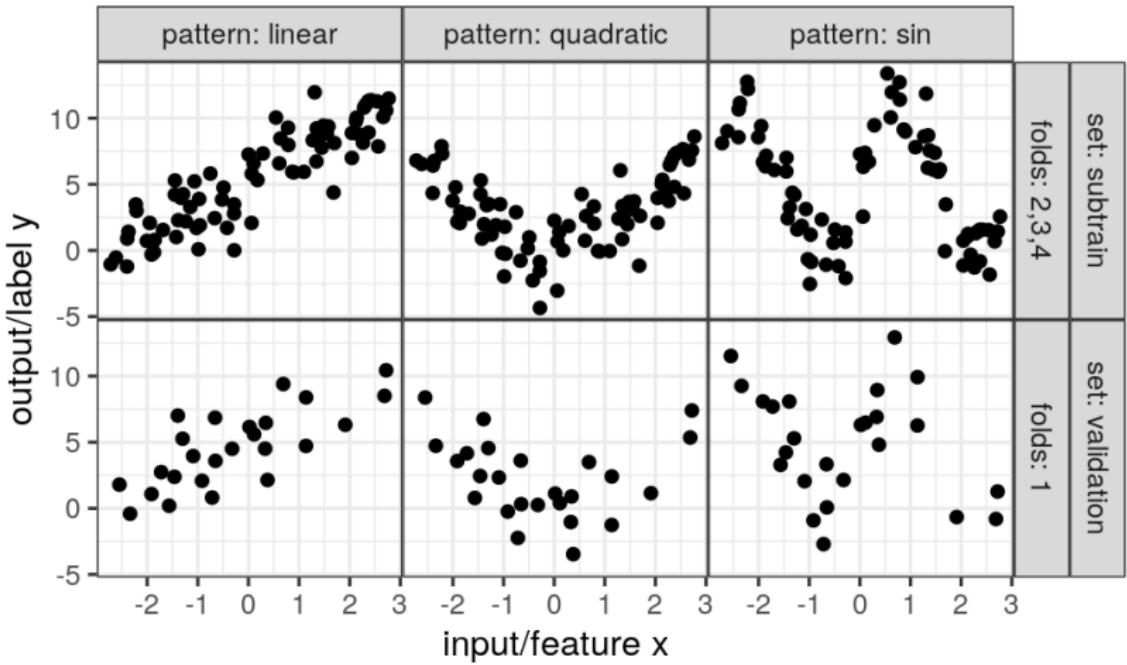
Practical considerations

Demonstration of overfitting in regression

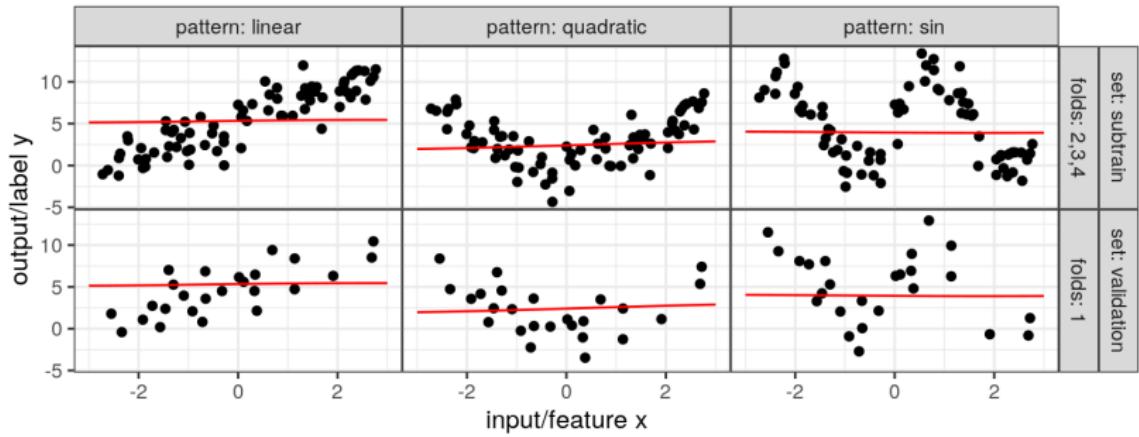
Classifying images of digits



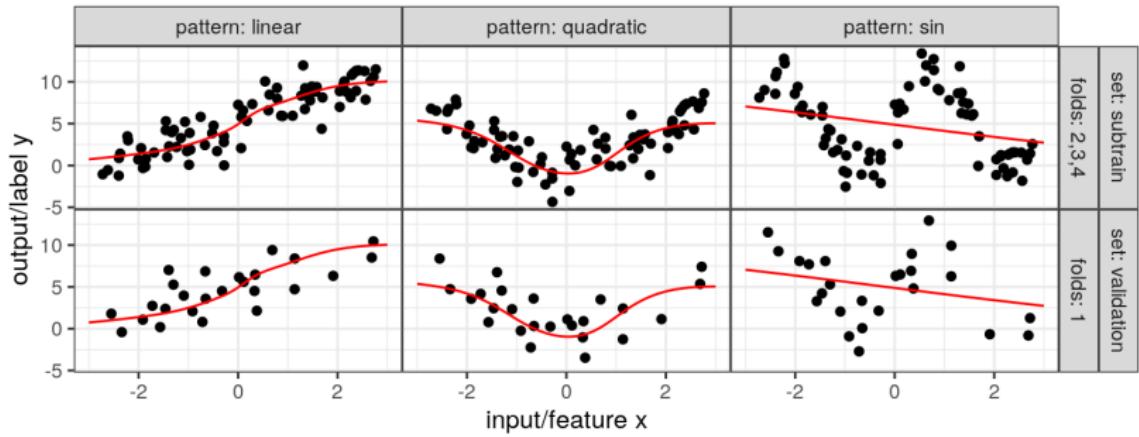




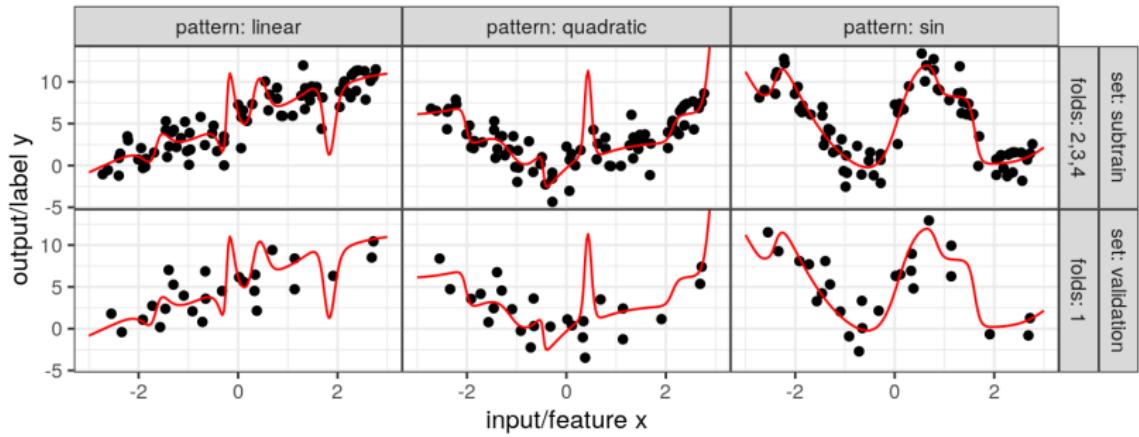
Neural network, 10 hidden units, 1 gradient descent iterations



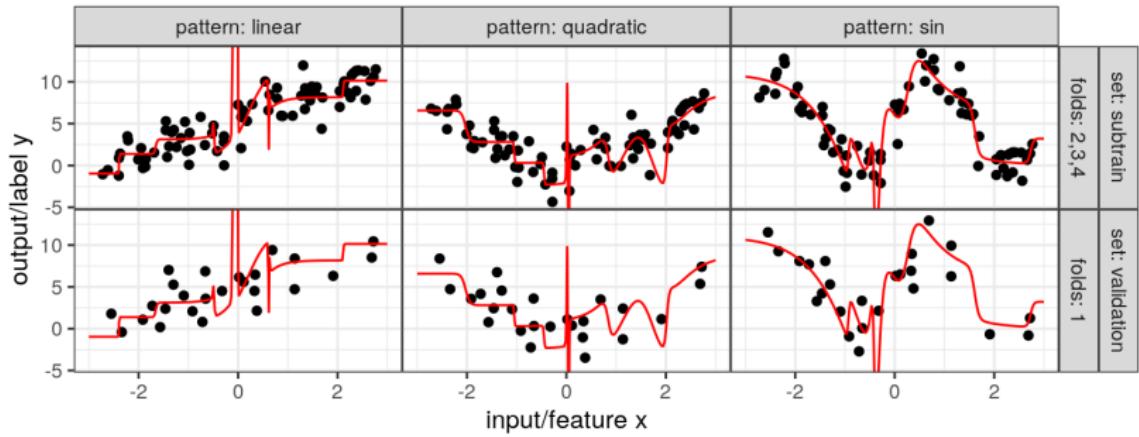
Neural network, 10 hidden units, 10 gradient descent iterations



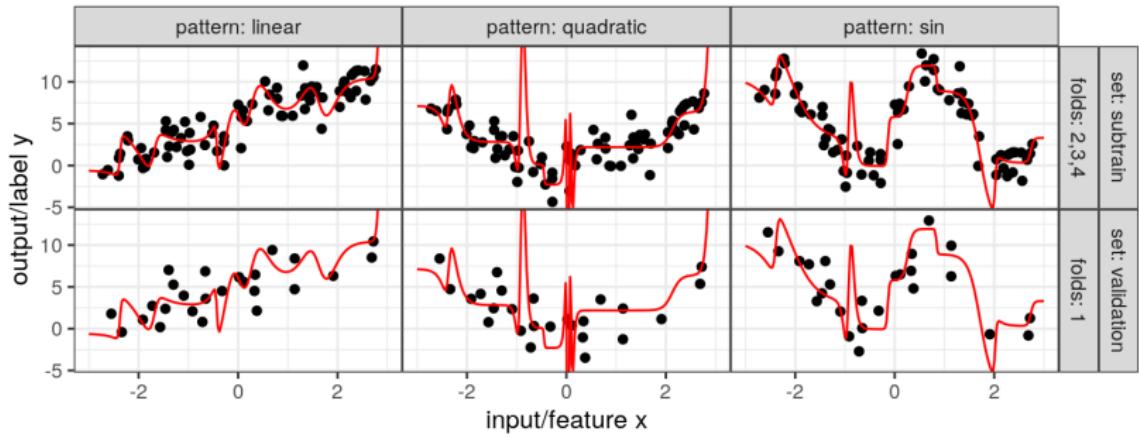
Neural network, 10 hidden units, 100 gradient descent iterations



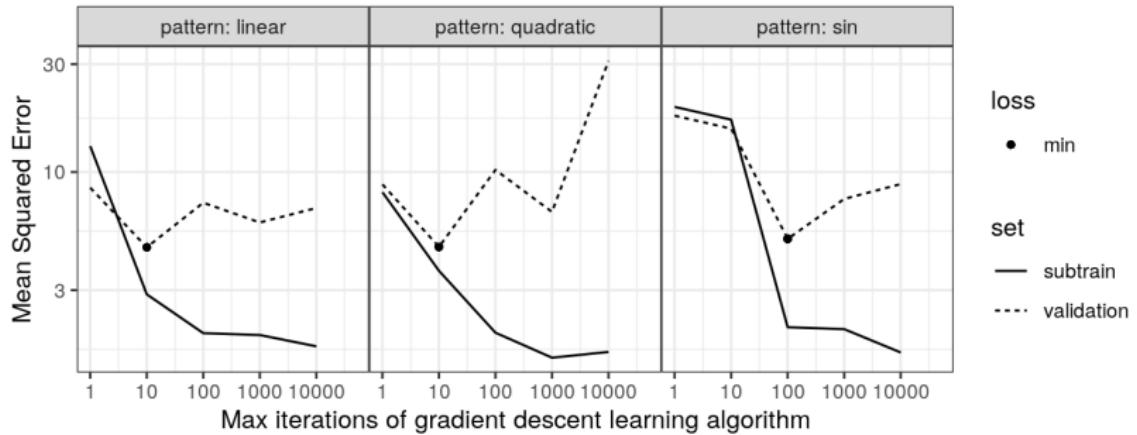
Neural network, 10 hidden units, 1000 gradient descent iterations



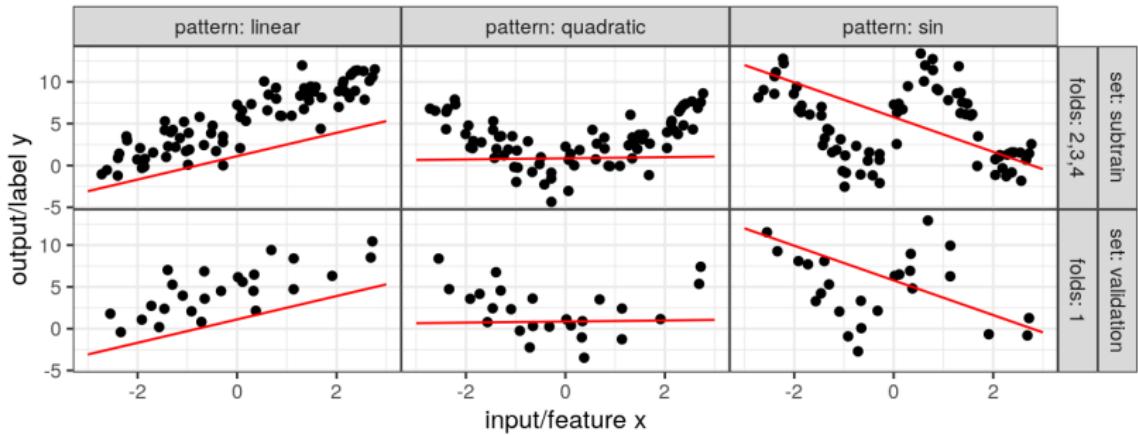
Neural network, 10 hidden units, 10000 gradient descent iterations



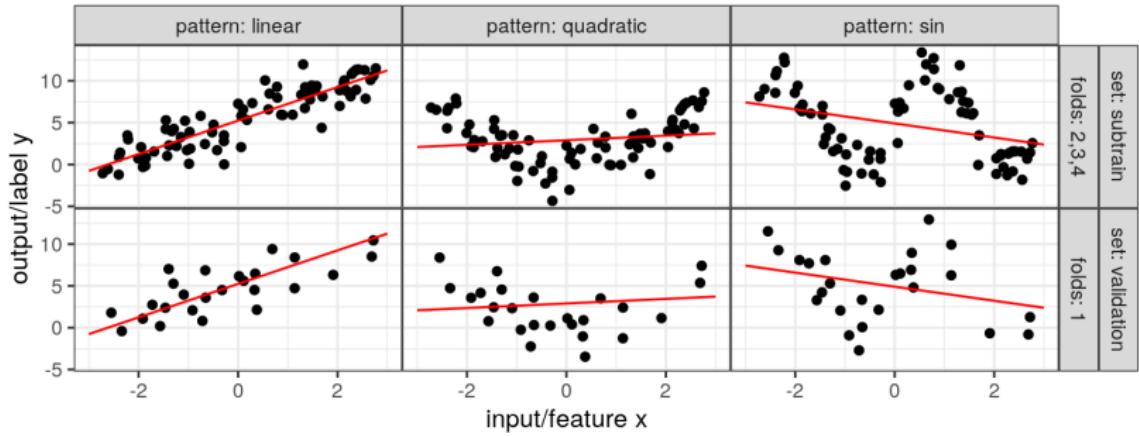
## Neural network, 10 hidden units



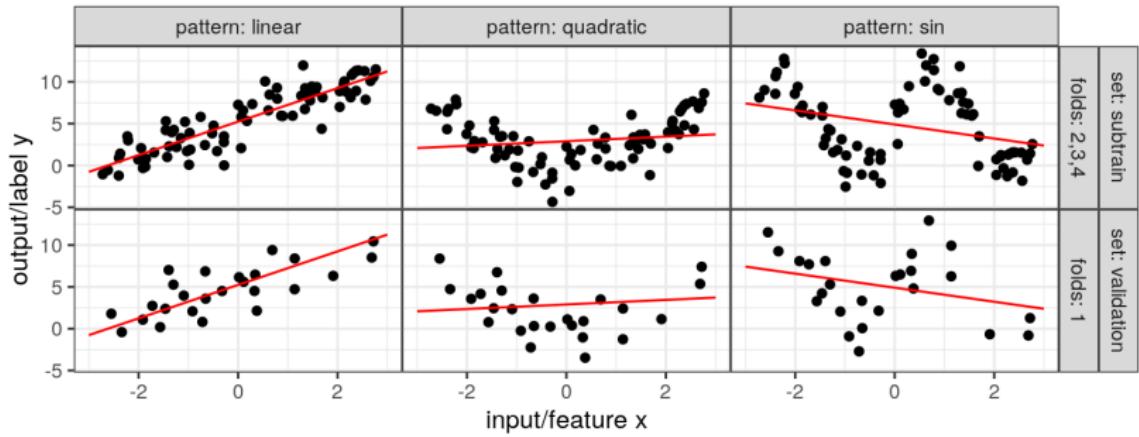
Neural network, 0 hidden units, 1 gradient descent iterations



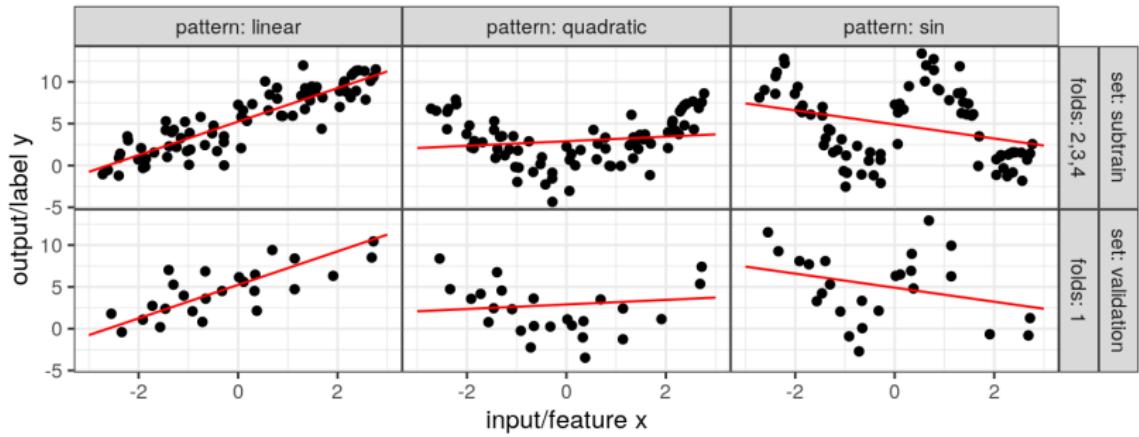
Neural network, 0 hidden units, 10 gradient descent iterations



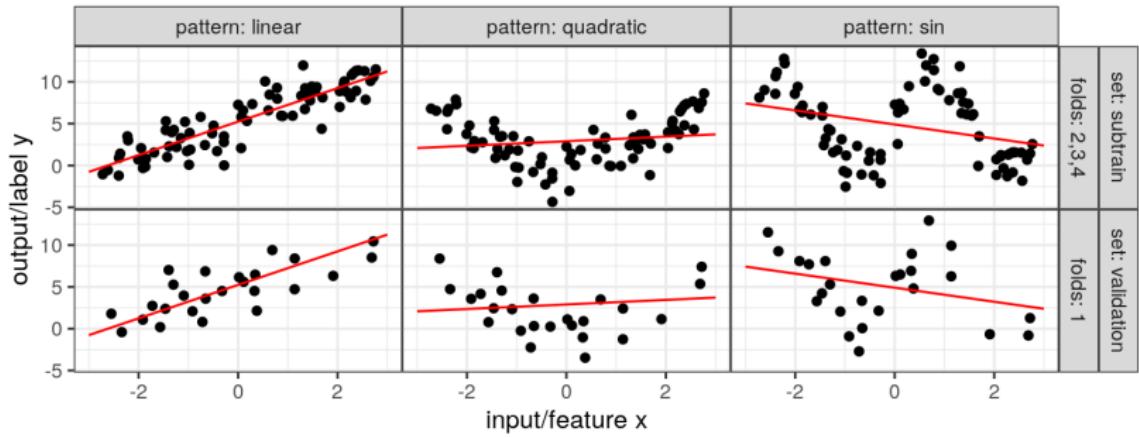
Neural network, 0 hidden units, 100 gradient descent iterations



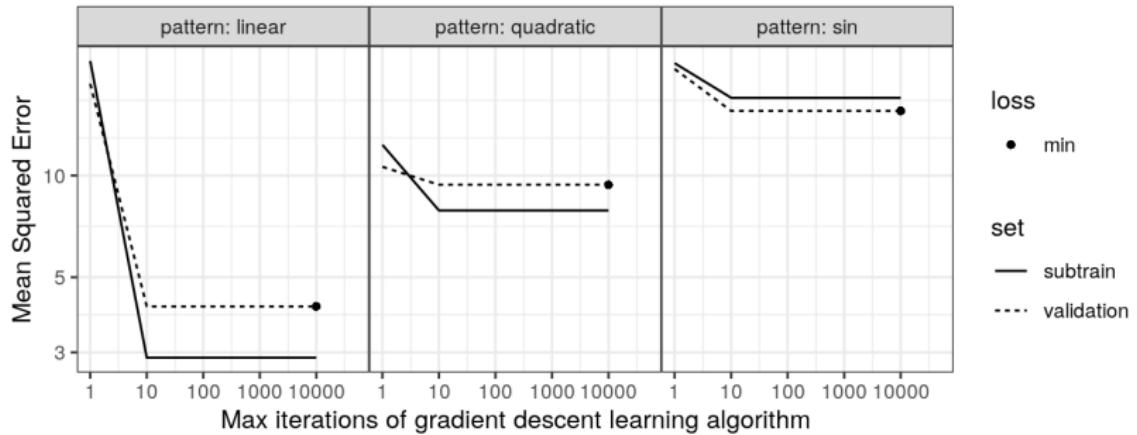
Neural network, 0 hidden units, 1000 gradient descent iterations



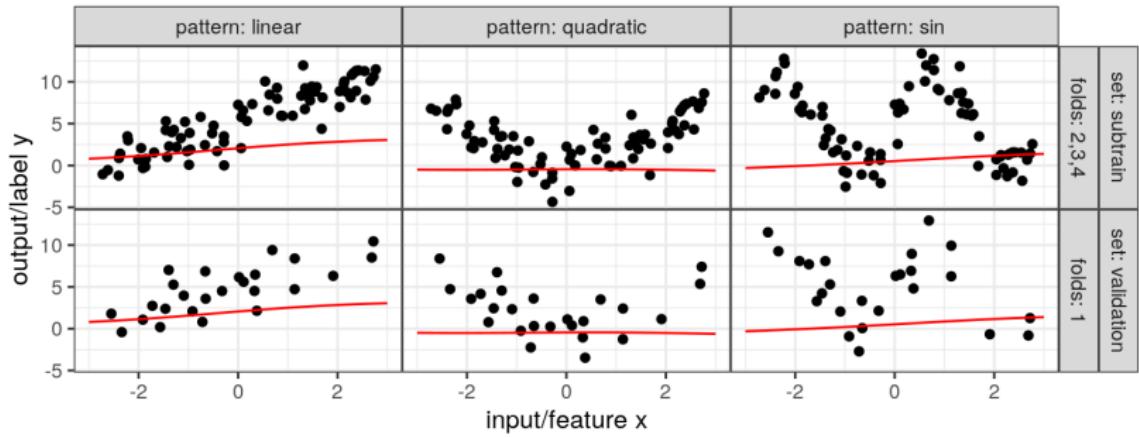
Neural network, 0 hidden units, 10000 gradient descent iterations



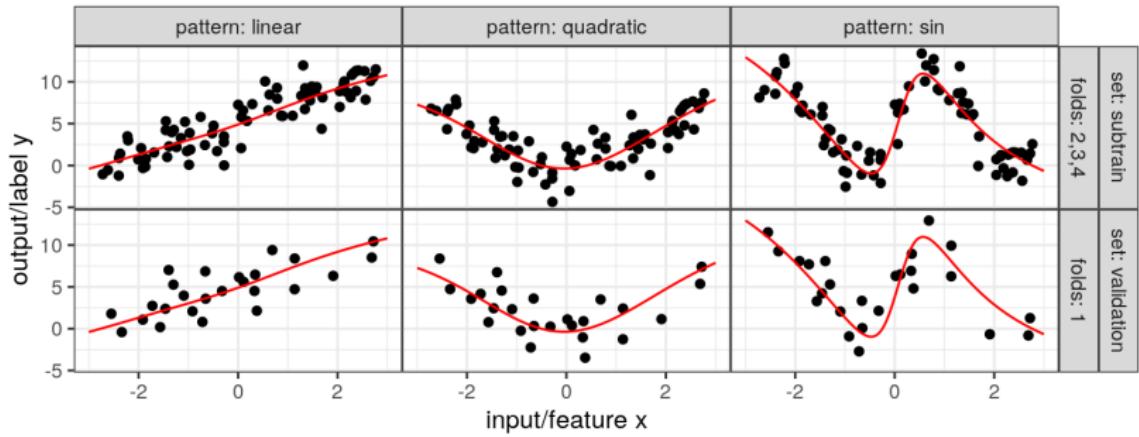
## Neural network, 0 hidden units



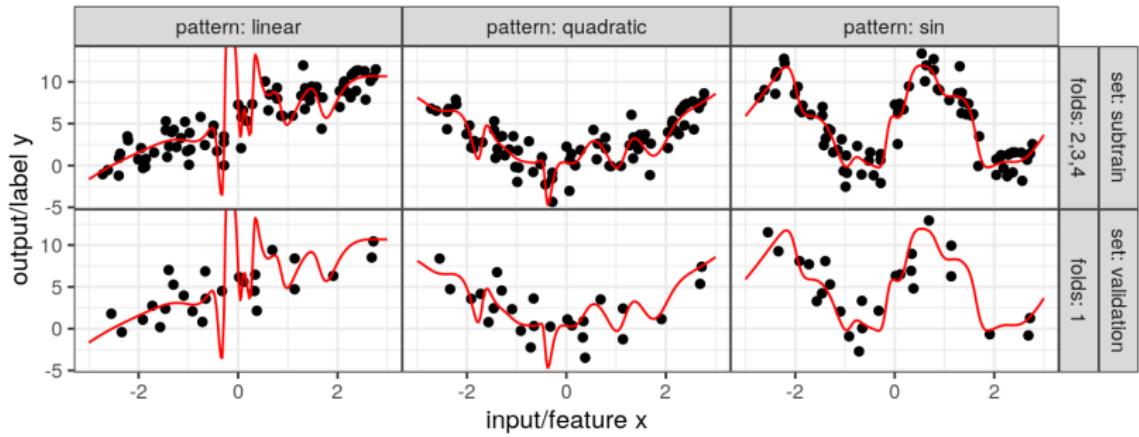
Neural network, 200 hidden units, 1 gradient descent iterations



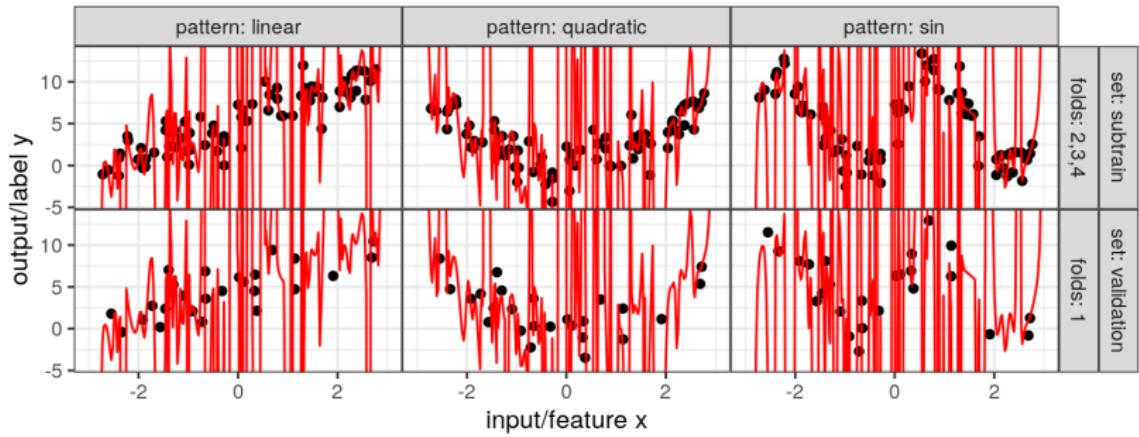
Neural network, 200 hidden units, 10 gradient descent iterations



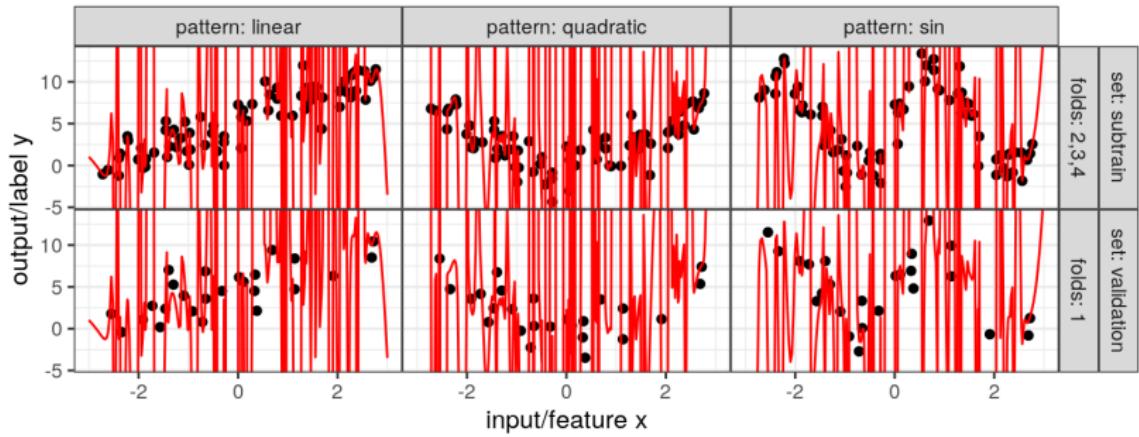
Neural network, 200 hidden units, 100 gradient descent iterations



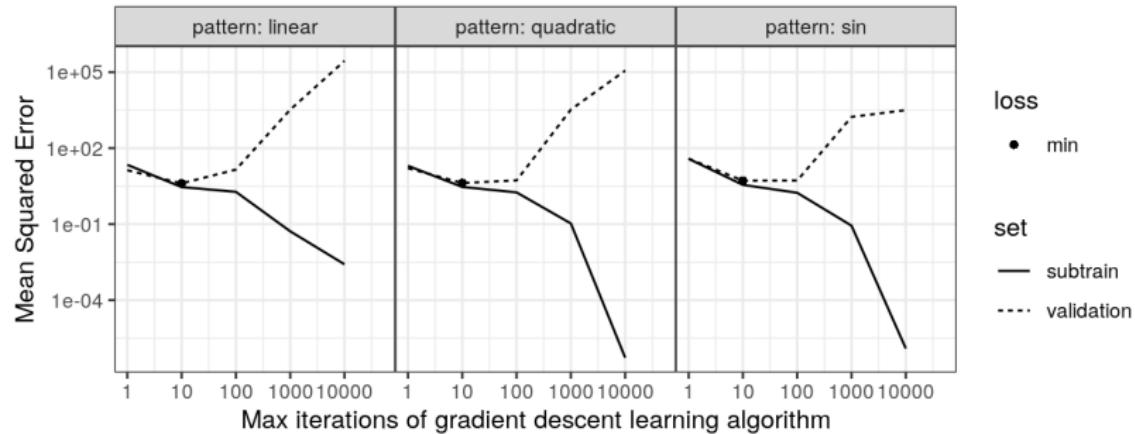
Neural network, 200 hidden units, 1000 gradient descent iterations



Neural network, 200 hidden units, 10000 gradient descent iterations



## Neural network, 200 hidden units

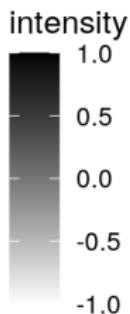
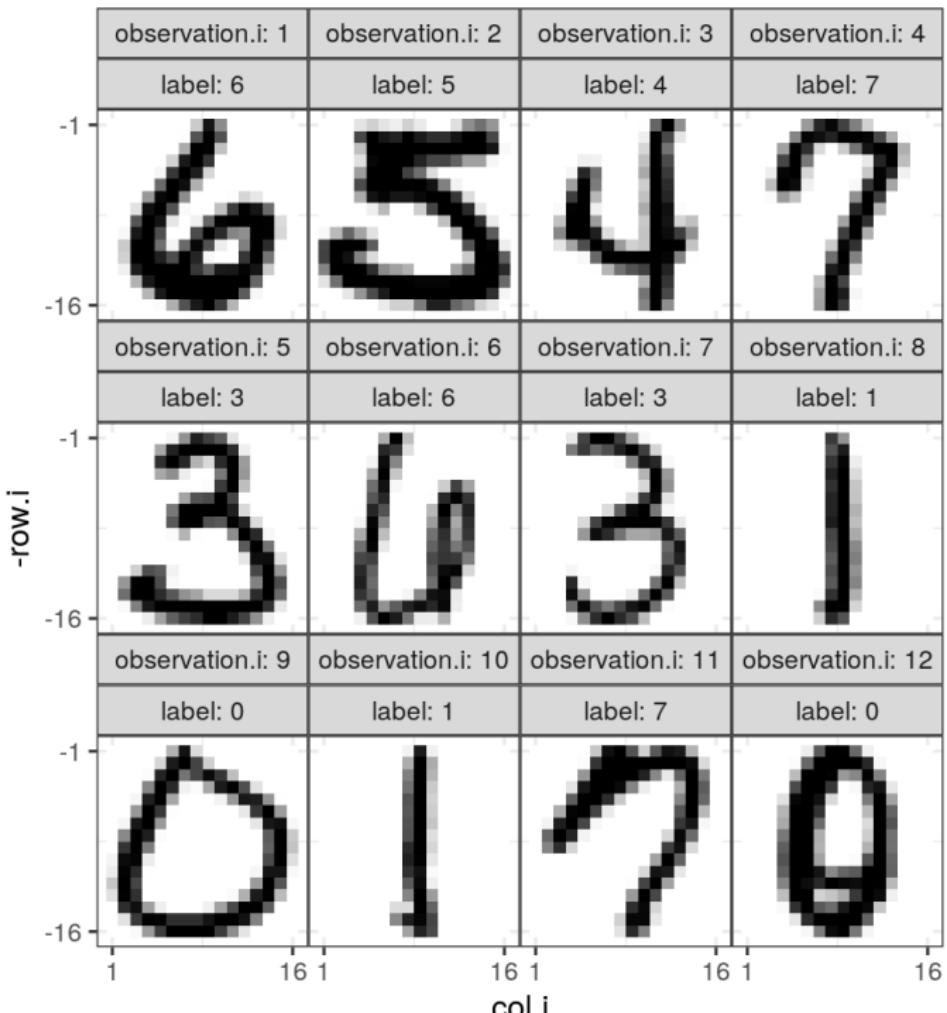


Introduction and applications

Practical considerations

Demonstration of overfitting in regression

Classifying images of digits



## Representation of digits in CSV

- ▶ Each image is one row.
- ▶ First row is output/label/class to predict.
- ▶ Other 256 columns are pixel intensity values.

1:	6	-1	-1	...	-1.000	-1.000	-1
2:	5	-1	-1	...	-0.671	-0.828	-1
3:	4	-1	-1	...	-1.000	-1.000	-1
4:	7	-1	-1	...	-1.000	-1.000	-1
5:	3	-1	-1	...	-0.883	-1.000	-1
6:	6	-1	-1	...	-1.000	-1.000	-1
7:	3	-1	-1	...	-1.000	-1.000	-1
8:	1	-1	-1	...	-1.000	-1.000	-1
9:	0	-1	-1	...	-1.000	-1.000	-1
10:	1	-1	-1	...	-1.000	-1.000	-1
11:	7	-1	-1	...	-1.000	-1.000	-1
12:	0	-1	-1	...	-1.000	-1.000	-1

## Converting label column to matrix for neural network

This is a “one hot” encoding of the class labels.

```
zip.dt <- data.table::fread("zip.gz")
zip.y.mat <- keras::to_categorical(zip.dt$V1)
```

	0	1	2	3	4	5	6	7	8	9
[1,]	0	0	0	0	0	0	1	0	0	0
[2,]	0	0	0	0	0	1	0	0	0	0
[3,]	0	0	0	0	1	0	0	0	0	0
[4,]	0	0	0	0	0	0	0	1	0	0
[5,]	0	0	0	1	0	0	0	0	0	0
[6,]	0	0	0	0	0	0	1	0	0	0
[7,]	0	0	0	1	0	0	0	0	0	0
[8,]	0	1	0	0	0	0	0	0	0	0
[9,]	1	0	0	0	0	0	0	0	0	0
[10,]	0	1	0	0	0	0	0	0	0	0
[11,]	0	0	0	0	0	0	0	1	0	0
[12,]	1	0	0	0	0	0	0	0	0	0

## Conversion to array for input to neural network

Use array function with all columns except first as data.

```
zip.size <- 16  
zip.X.array <- array(  
  data = unlist(zip.dt[1:nrow(zip.dt), -1]),  
  dim = c(nrow(zip.dt), zip.size, zip.size, 1))
```

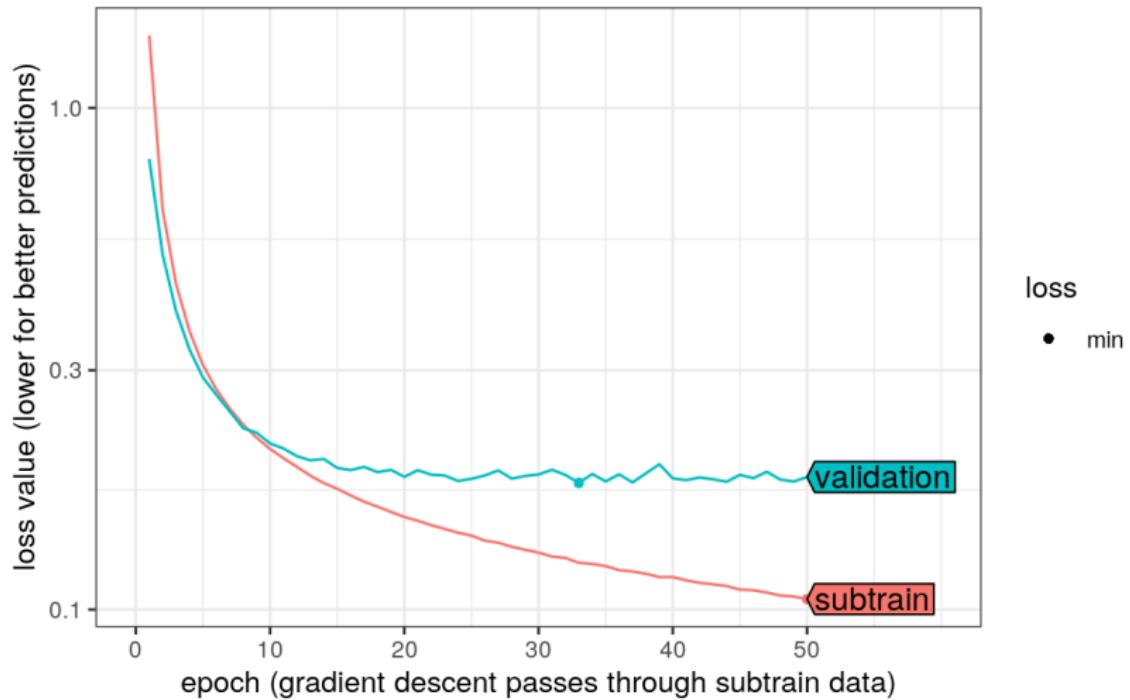
Need to specify dimensions of array:

- ▶ Observations: same as the number of rows in the CSV table.
- ▶ Pixels wide: 16.
- ▶ Pixels high: 16.
- ▶ Channels: 1 (greyscale image).

## Linear model R code

```
library(keras)
linear.model <- keras_model_sequential() %>%
  layer_flatten(
    input_shape = dim(zip.X.array)[-1]) %>%
  layer_dense(
    units = ncol(zip.y.mat),
    activation = 'softmax')
```

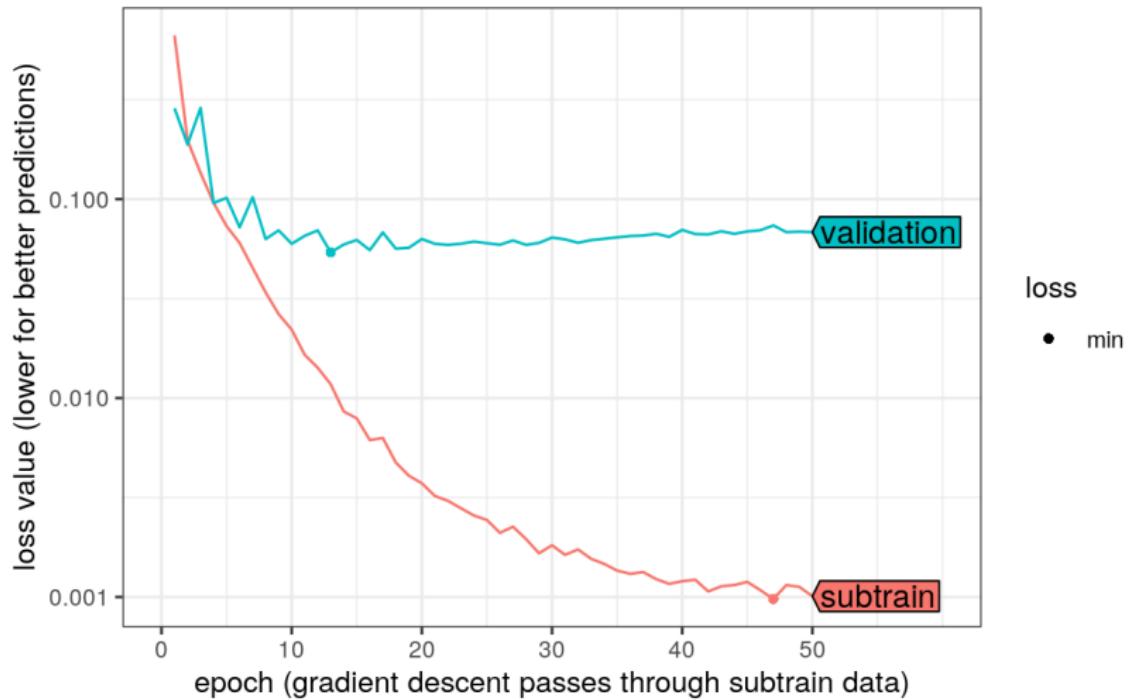
## Linear model



## Convolutional model R code

```
library(keras)
conv.model <- keras_model_sequential() %>%
  layer_conv_2d(
    input_shape = dim(zip.X.array)[-1],
    filters = 20,
    kernel_size = c(3,3),
    activation = 'relu') %>%
  layer_max_pooling_2d(pool_size = c(2, 2)) %>%
  layer_flatten() %>%
  layer_dense(units = 100, activation = 'relu') %>%
  layer_dense(
    units = ncol(zip.y.mat),
    activation = 'softmax')
```

## Convolutional neural network



## Dense (fully connected) neural network R code

```
library(keras)
dense.model <- keras_model_sequential() %>%
  layer_flatten(
    input_shape = dim(zip.X.array)[-1]) %>%
  layer_dense(units = 100, activation = 'relu') %>%
  layer_dense(
    units = ncol(zip.y.mat),
    activation = 'softmax')
```

## Dense (fully connected) neural network with 8 hidden layers

