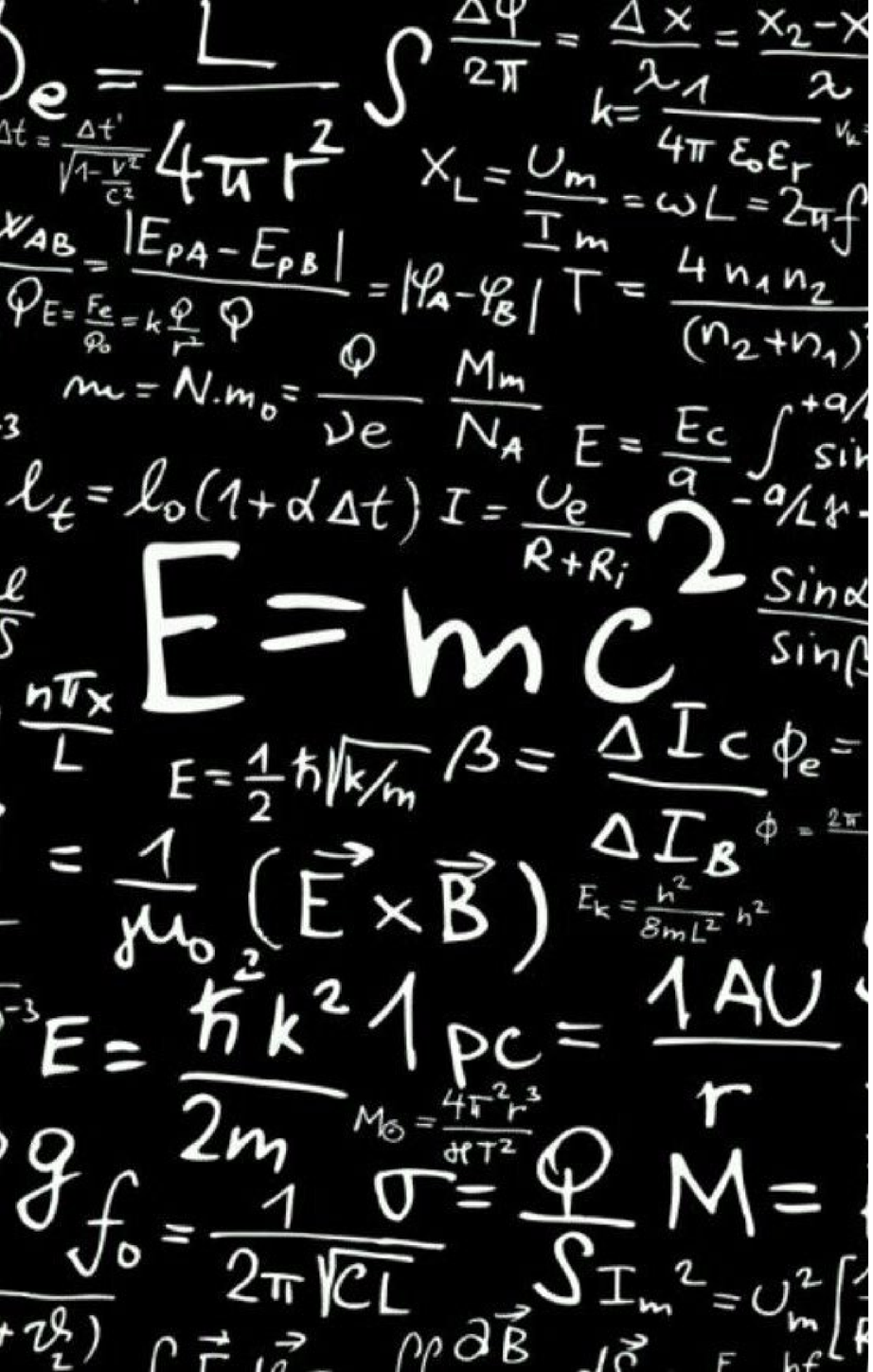

HYPERPARAMETER TUNING

Parameters

- Variables in the model
- Values are adjusted based on the given data

Parameters

- Variables in the model
- Values are adjusted based on the given data

$$h(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3$$

- w_0, w_1, w_2, w_3 are parameters (weights) of the model
- These variables are “tuned”/ “learn” automatically during the training

Other Parameters

- However, given the same inputs, there are other factors that can affect the result of the model.
- Let’s look at a neural network. What else can affect the performance?

Other Parameters

- However, given the same inputs, there are other factors that can affect the result of the model.
- Let’s look at a neural network. What else can affect the performance?

  Model structure

  Initial state

  Learning rate

  Optimizer

  Activation function

  Regularization

  Max iterations

  …

Hyperparameters

These parameters are used to control the learning of the model. We call it hyperparameters.

Model structure	how does the model look like?
Initial state	what are the initial values of parameters?
Learning rate	how fast should the model learn?
Optimizer	how to optimize the learning process?
Activation function	what non-linearity should the model use?
Regularization	how much should the model regularize?
Max iterations	how long should the model learn?
…	

Validation Set

How can we know what hyperparameter set is the best?

- Should we use training set to test the performance?

Validation Set

How can we know what hyperparameter set is the best?

- Should we use training set to test the performance?
- How about the test set?

Validation Set

This is yet another reason why we need to split a model into 3 parts: training set, validation set, test set.

- We train using training set. (parameter optimization)
- We validate the hyperparameters using validation set. (hyperparameter optimization)
- We access the model performance using test set

https://www.jeremyjordan.me/hyperparameter-tuning/

Best hyperparameters?

How do we find the best set of hyperparameters?

Best hyperparameters

How do we find the best set of hyperparameters?

- We search. (Grid search, random search, …)

```
# get some data
X, y = load_digits(return_X_y=True)

# build a classifier
clf = SGDClassifier(loss='hinge', penalty='elasticnet',
                    fit_intercept=True)

param_grid = {'average': [True, False],
              'l1_ratio': np.linspace(0, 1, num=10),
              'alpha': np.power(10, np.arange(-4, 1, dtype=float))}

# run grid search
grid_search = GridSearchCV(clf, param_grid=param_grid)
start = time()
grid_search.fit(X, y)
```

Load in dataset

Best hyperparameters

How do we find the best set of hyperparameters?

- We search. (Grid search, random search, …)

```
# get some data
X, y = load_digits(return_X_y=True)

# build a classifier
clf = SGDClassifier(loss='hinge', penalty='elasticnet',
                    fit_intercept=True)

param_grid = {'average': [True, False],
              'l1_ratio': np.linspace(0, 1, num=10),
              'alpha': np.power(10, np.arange(-4, 1, dtype=float))}

# run grid search
grid_search = GridSearchCV(clf, param_grid=param_grid)
start = time()
grid_search.fit(X, y)
```

Specify the model

Best hyperparameters

How do we find the best set of hyperparameters?

- We search. (Grid search, random search, …)

```
# get some data
X, y = load_digits(return_X_y=True)

# build a classifier
clf = SGDClassifier(loss='hinge', penalty='elasticnet',
                    fit_intercept=True)

param_grid = {'average': [True, False],
              'l1_ratio': np.linspace(0, 1, num=10),
              'alpha': np.power(10, np.arange(-4, 1, dtype=float))}

# run grid search
grid_search = GridSearchCV(clf, param_grid=param_grid)
start = time()
grid_search.fit(X, y)
```

Set up the dictionary of hyperparameters

Best hyperparameters

How do we find the best set of hyperparameters?

- We search. (Grid search, random search, …)

```
# get some data
X, y = load_digits(return_X_y=True)

# build a classifier
clf = SGDClassifier(loss='hinge', penalty='elasticnet',
                    fit_intercept=True)

param_grid = {'average': [True, False],
              'l1_ratio': np.linspace(0, 1, num=10),
              'alpha': np.power(10, np.arange(-4, 1, dtype=float))}
```

```
# run grid search
grid_search = GridSearchCV(clf, param_grid=param_grid)
start = time()
grid_search.fit(X, y)
```

Run the search. Best hyperparameters set will be stored in best_estimator_ variable.