Hyper-parameter search

git pull main and run first few cells :)

Divyam Madaan

Building a classifier

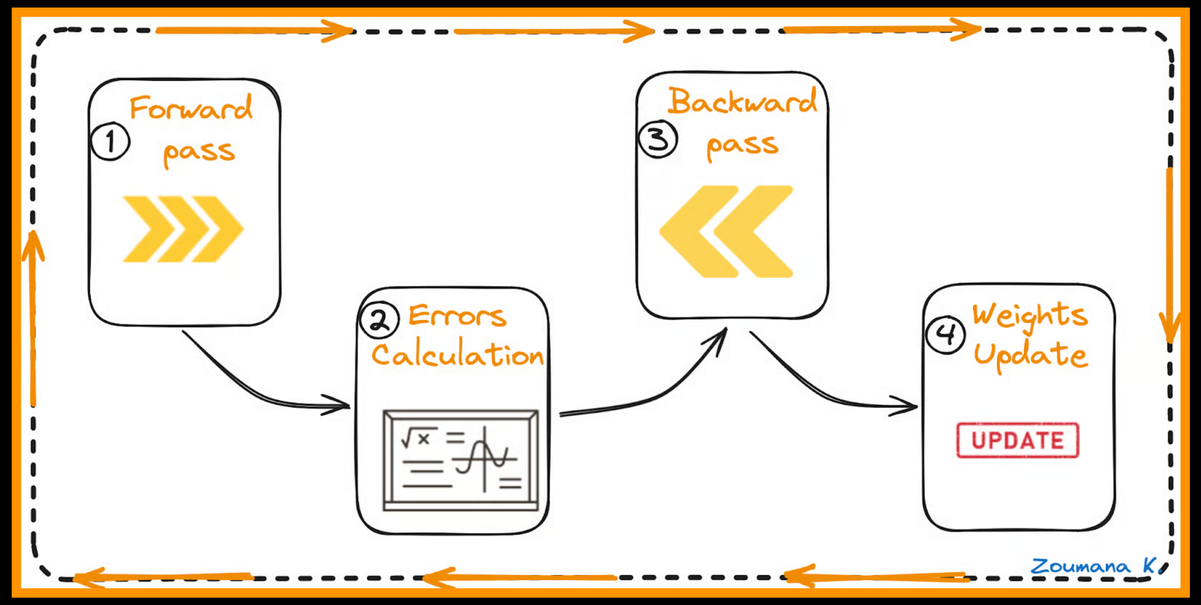


Step 1: Load your dataset

Step 2: Train your model

Step 3: Evaluate your model

How do we evaluate our models?

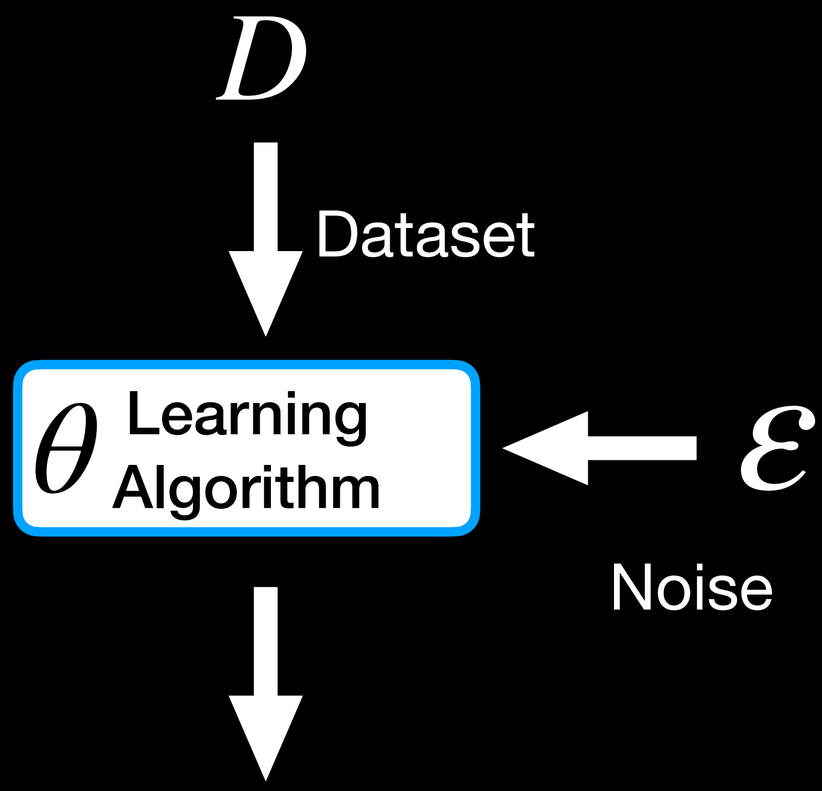



```
model = Model( )
criterion =nn.CrossEntropyLoss( )
optimizer = optim.Adam(net.parameters(), lr=lr)
for epoch in range(epochs):
    for i, data in enumerate(train_loader):
        Training loop
```

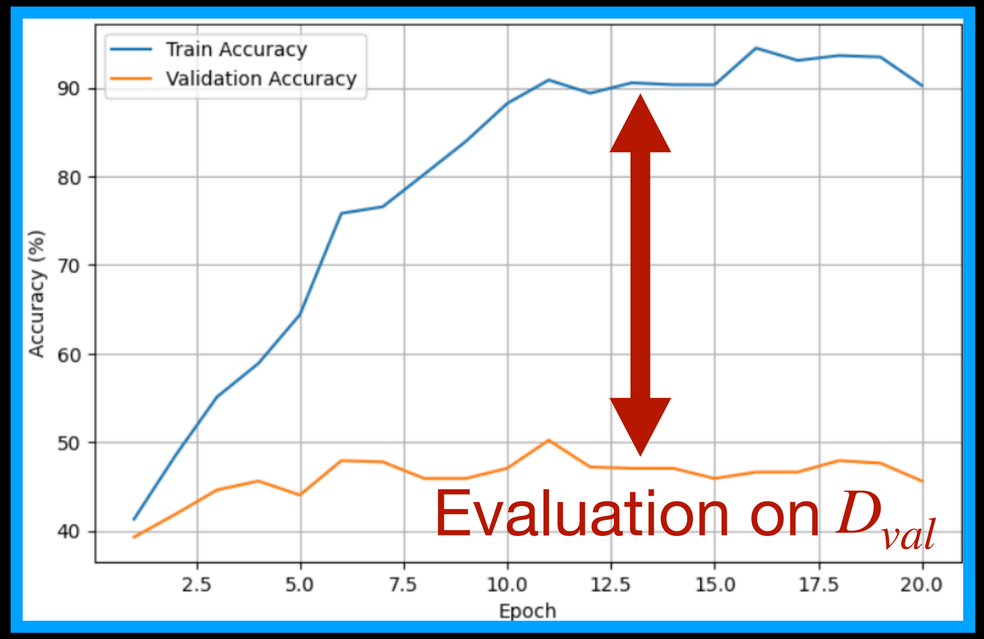
Objectives

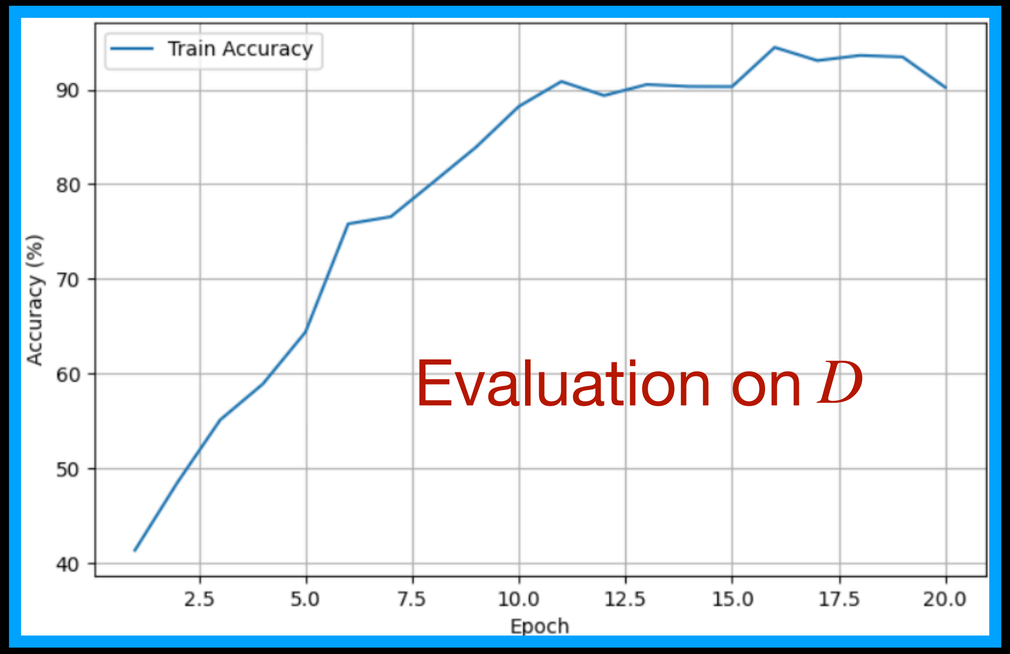
- Cross-validation
- Hyper-parameter search
    - Random Search
    - Bayesian Search

Learning and evaluation



Model evaluation

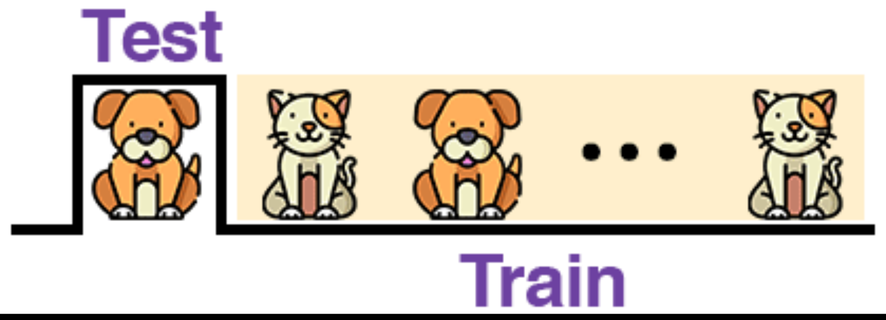


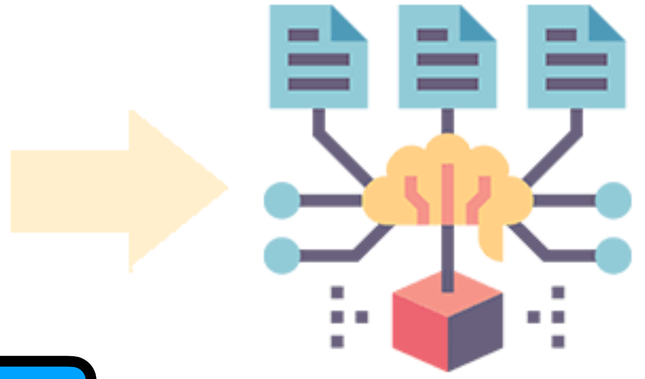


Cross validation

$$\hat{R}^{(CV)} := \mathbb{E}_{D_1,\ldots,D_N}\left[\frac{1}{N}\sum_{n=1}^{N}\hat{R}(\mathrm{Learn}(D_{-n};\varepsilon),D_n)\right].$$

Data: $D_n(n=1,\ldots N); D=\cup_{n=1}^{N} D_n$ **Model:**



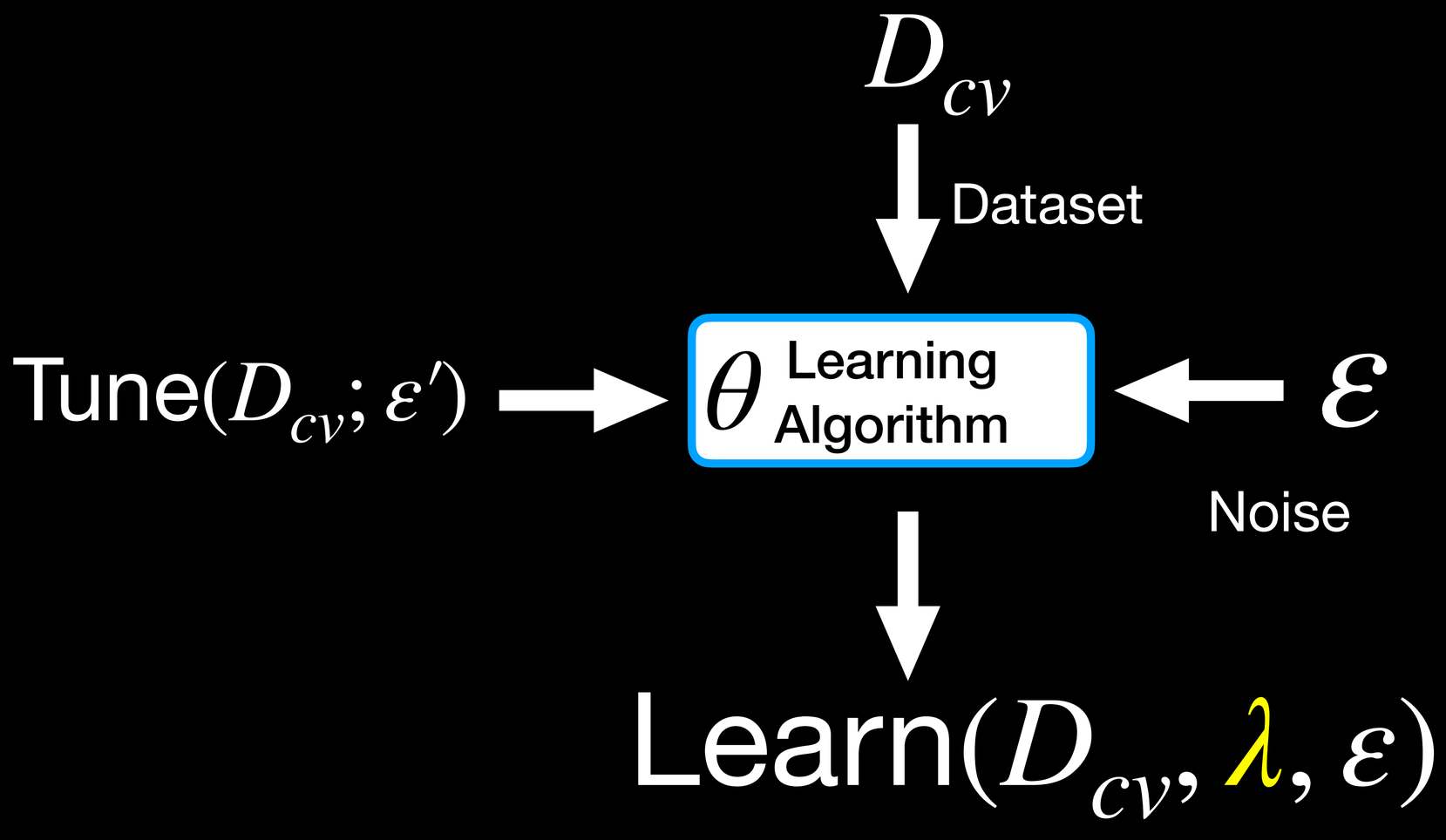


Train-set variation or test-set variation?

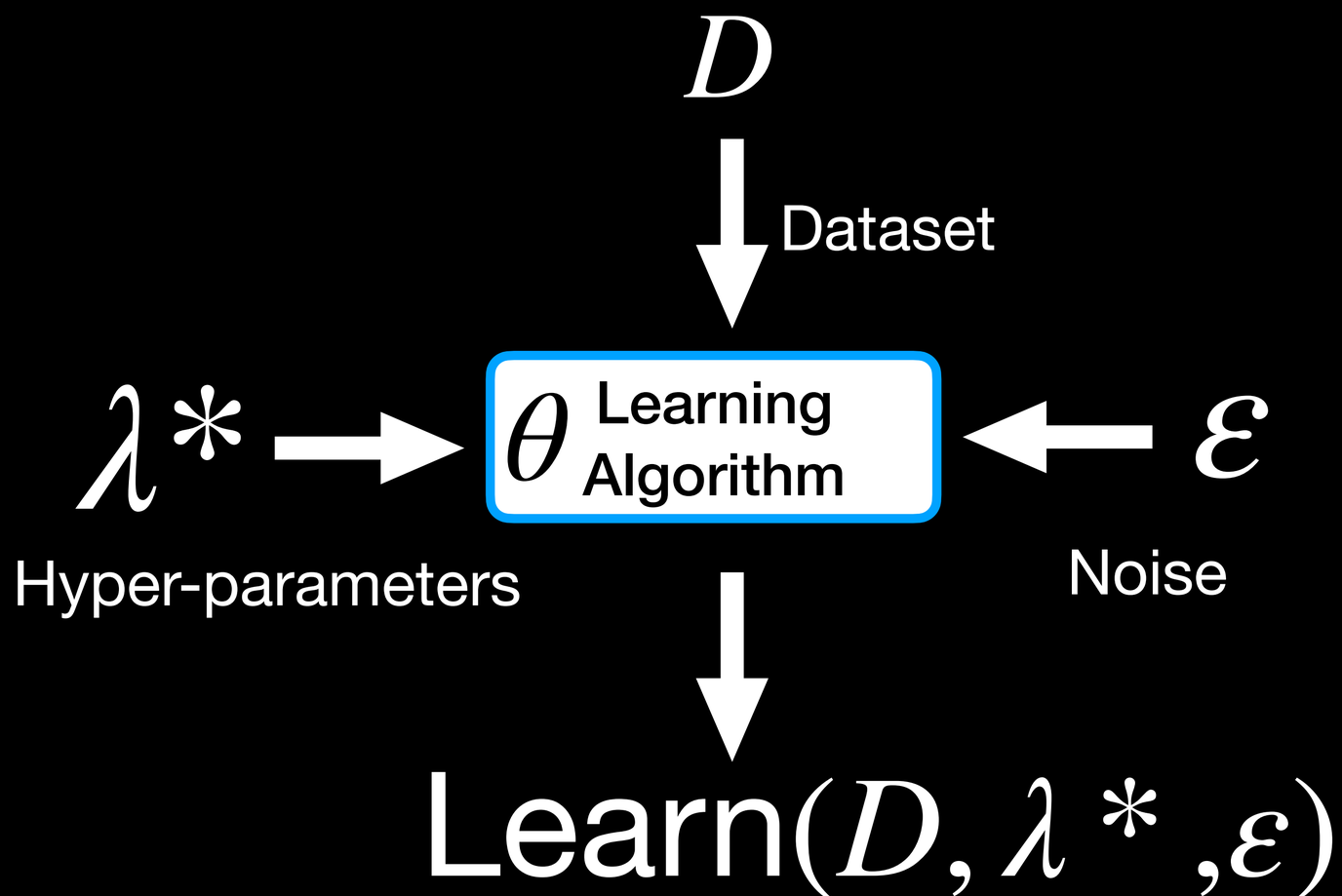
Objectives

- Cross-validation
- Hyper-parameter search
    - Random Search
    - Bayesian Search

Hyper-parameter search



Hyper-parameter search



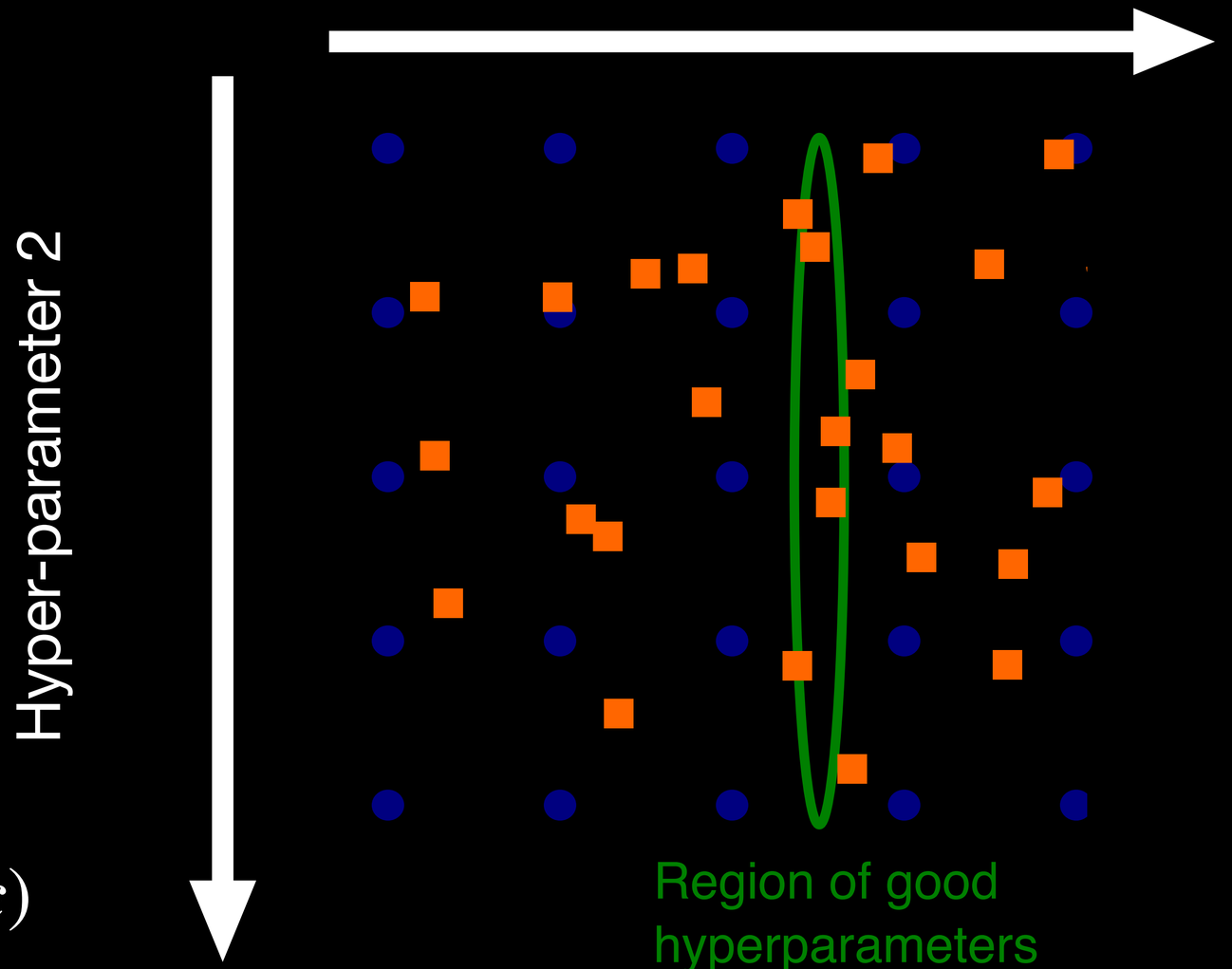
Random search

$p(\lambda)$

Prior distribution
over hyper-parameters

$\{\lambda_1, \ldots, \lambda_k\}$

$$\lambda^* = \arg \min_{\lambda \in \{\lambda_i\}_{i=1}^K} \hat{R}(\text{Learn}(D_{cv}; \lambda, \varepsilon)$$



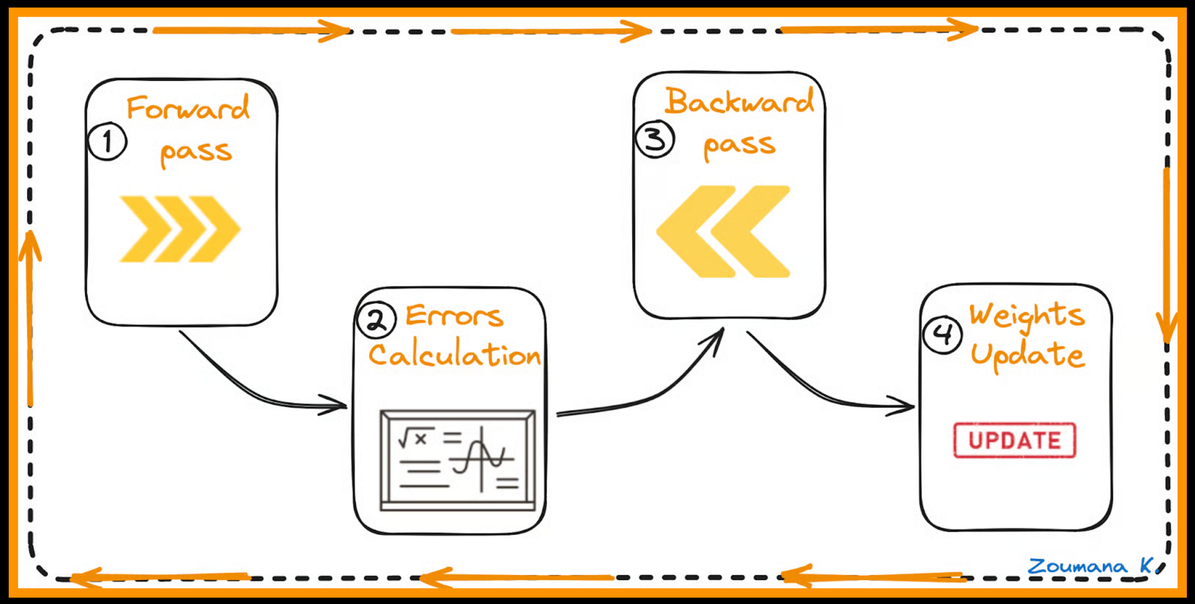
Building a classifier



Step 1: Load your dataset

Step 2: Train your model

Step 3: Evaluate your model




```
model = Model( )
criterion =nn.CrossEntropyLoss( )
optimizer = optim.Adam(net.parameters(), lr=lr)
for epoch in range(epochs):
    for i, data in enumerate(train_loader):
        Training loop
```

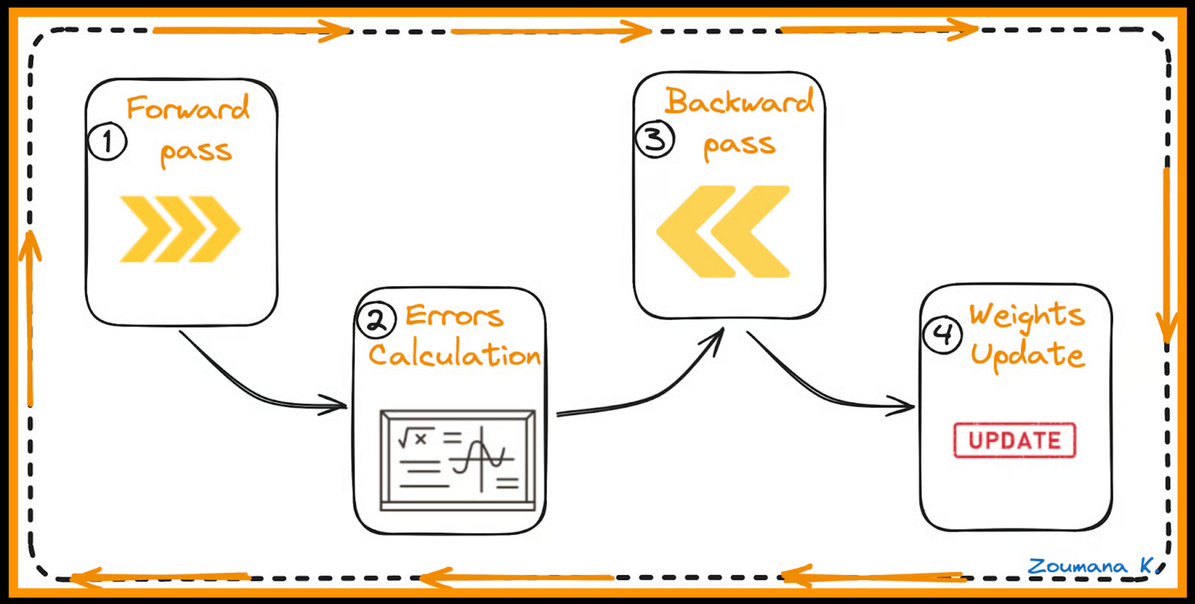
Building a classifier



Step 1: Load your dataset

Step 2: Train your model

Step 3: Evaluate your model

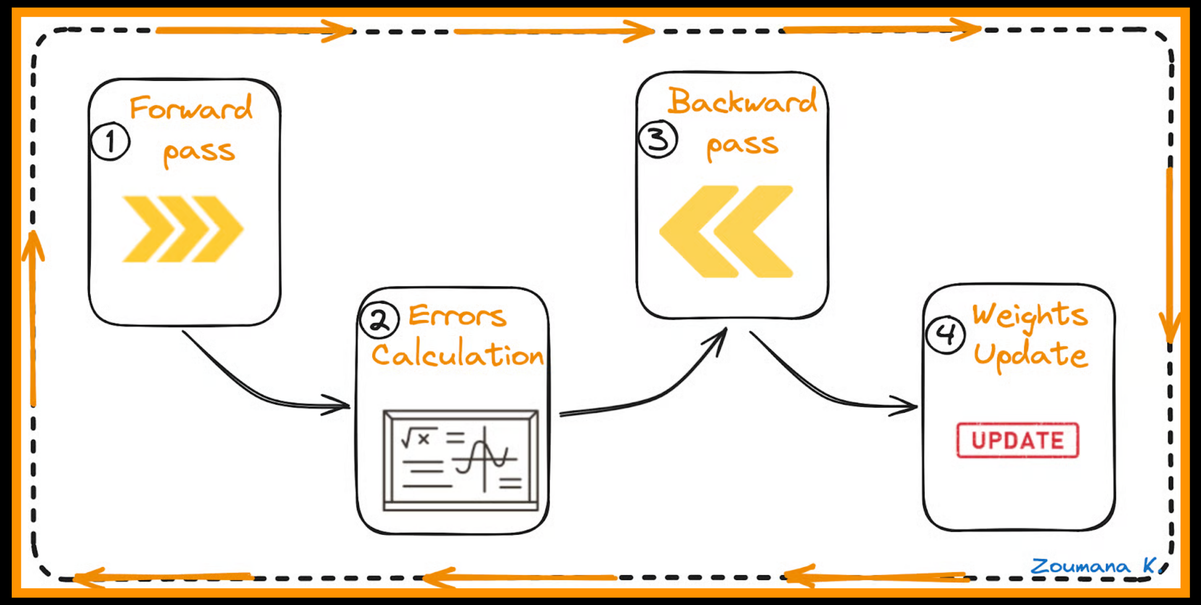



```
model = Model( )
net = NeuralNetClassifier(
                module=model,
                optimizer=optim.SGD,
                max_epochs=10,
                lr=1e-3,
                device=device)

net.fit(X_train, Y_train)
```

Random Search

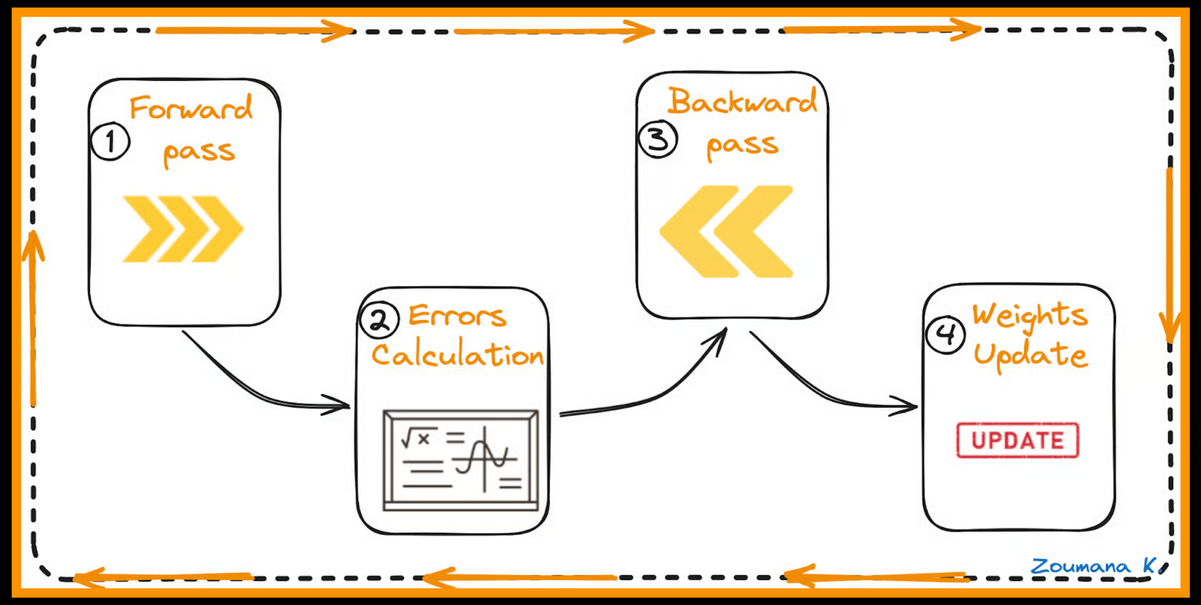
```
model = Net( )
net = NeuralNetClassifier(
            module=model,
            optimizer=optim.SGD,
            max_epochs=10,
            lr=1e-3,
            device=device)
random_search = RandomizedSearchCV(
                    estimator=__,
                    param_distributions,
                    ___,___,
                    scoring="accuracy")
random_search.fit(X_train, Y_train)
```




```
model = Model( )
net = NeuralNetClassifier(
            module=model,
            optimizer=optim.SGD,
            max_epochs=10,
            lr=1e-3,
            device=device)
net.fit(X_train, Y_train)
```

Bayesian Search

```
model = Net( )
net = NeuralNetClassifier(
            module=model,
            optimizer=optim.SGD,
            max_epochs=10,
            lr=1e-3,
            device=device)
bayes_search = BayesSearchCV(
                    estimator=net,
                    param_distributions,
                    n_iter=10,cv=3,
                    scoring="accuracy")
bayes_search.fit(X_train, Y_train)
```




```
model = Model( )
net = NeuralNetClassifier(
            module=model,
            optimizer=optim.SGD,
            max_epochs=10,
            lr=1e-3,
            device=device)
net.fit(X_train, Y_train)
```