Training Process

1. Initialize the model weights randomly
2. Predict a few examples with the current weights
3. Compare prediction with true labels
4. Calc how to change weights to improve predictions
5. Update weights slightly
6. Go back to 2.

https://course.spacy.io/en/chapter4

Generate a Configuration File for Training

`python -m spacy init config ./conf ig.cfg --lang en --pipeline ner`

This will allow training for the ner pipeline

init config: the command to run
config.cfg: output path for the generated config
--lang: language class of the pipeline, e.g. en for English
--pipeline: comma-separated names of components to include

Create Training Data (with DocBin)

```
from spacy.t okens import DocBin
```

Create and save a collection of training docs

```
docs train_ docbin = DocBin (do cs= tra in_ docs)
train_docbin.to_disk("./train.spacy")
```

Create and save a collection of evaluation docs

```
dev_docbin = DocBin (do cs= dev _docs)
dev_docbin.to_disk("./dev.spacy")
```

(via Sypder or Jupyter using DocBin)

Training the Data with CLI

if used a base_config.cfg file

```
python -m spacy init fill-c onfig base_c onf -
ig.cfg config.cfg
```

if configurations entered in config.cfg (namely the dev/train paths)

```
python -m spacy train config.cfg --output ./output
```

overwrite config file and train

```
python -m spacy train ./conf ig.cfg --output
./output --path s.train train.s pacy --path s.dev
dev.spacy
```

other way to overwrite config file settings (ex.)

in config file:
[training] `--training`
eval_frequency `.eval_ fre quency 10`
max_steps `.max_steps 300`

config file to cmd line:

```
python -m spacy train config.cfg --output ./output
--trai nin g.e val _fr equency 10 --trai nin g.m -
ax_ steps 300
```

https://spacy.io/usage/training

Train from Python Compiler

```
from spacy.c li.train import train as spacy_ train
config _path = "./c onf ig/ con fig.cf g"
output_model_path = " out put /"
spacy_train(
    config_path,
    output_path=output_model_path,
    overrides={
        "paths.train": "./t rai n.s pac y",
        "paths.dev": "./t est.sp acy ",
        "training.eval_frequency" : 10,
        "training.max_steps" : 300
    },
)
```

Train from Python Compiler (cont)

output:

ℹ Saving to output directory: output\
ℹ Using CPU
ℹ To switch to GPU 0, use the option: --gpu-id 0

```
========= Initia lizing pipeline ========
✔ Initia lized pipeline
====== ===== Training pipeline ==========
```

ℹ Pipeline: ['tok2vec', 'ner']
ℹ Initial learn rate: 0.001

```
E # LOSSTO K2VEC LOSSNER ENTS_F     ENT S_P     -
ENT S_R     SCORE
--- --- --- ---- ---- ---- ------- ------ ---- --
- ---
0    0   0.00       69.09       13.42       10.0 -
9   2 0.0 0   0.13
0   10       0.9 6   855.31      3.59       42.8 -
6   1.88      0.04
...
```

(etc)

✔ Saved pipeline to output directory
output \mo del -last

Trainable Components

tagger	morphologizer	trainable_lemmatizer
parser	ner	
spancat	texcat	

Configuration File (Defaults - sample)

```
python -m spacy init config ./conf ig.cfg --lang
en --pipeline ner
```

[paths]
train = null
dev = null
vectors = null
init_tok2vec = null

[nlp]
lang = "en"
pipeline = ["tok2vec","ner"]
batch_size = 1000
disabled = []
before_creation = null
after_creation = null
after_pipeline_creation = null
tokenizer = {"@tokenizers":"spacy.Tokenizer.v1"}

Configuration File (Defaults - sample) (cont)

[training]
dev_corpus = "corpora.dev"
train_corpus = "corpora.train"
seed = ${system.seed}
gpu_allocator = ${system.gpu_allocator}
dropout = 0.1
accumulate_gradient = 1
patience = 1600
max_epochs = 0
max_steps = 20000
eval_frequency = 200
frozen_components = []
annotating_components = []
before_to_disk = null

[training.batcher.size]
@schedules = "compounding.v1"
start = 100
stop = 1000
compound = 1.001
t = 0.0

[pretraining]

[initialize]
vectors = ${paths.vectors}
init_tok2vec = ${paths.init_tok2vec}

[initialize.components]
[initialize.tokenizer]

enter in path for train.spacy and test.spacy in train and dev for [paths] respectively
enter in trained pipeline in vectors for [path]
custom rules initialized near bottom

config file with annotations:
https://github.com/explosion/spaCy/blob/master/spacy/default_config.cfg