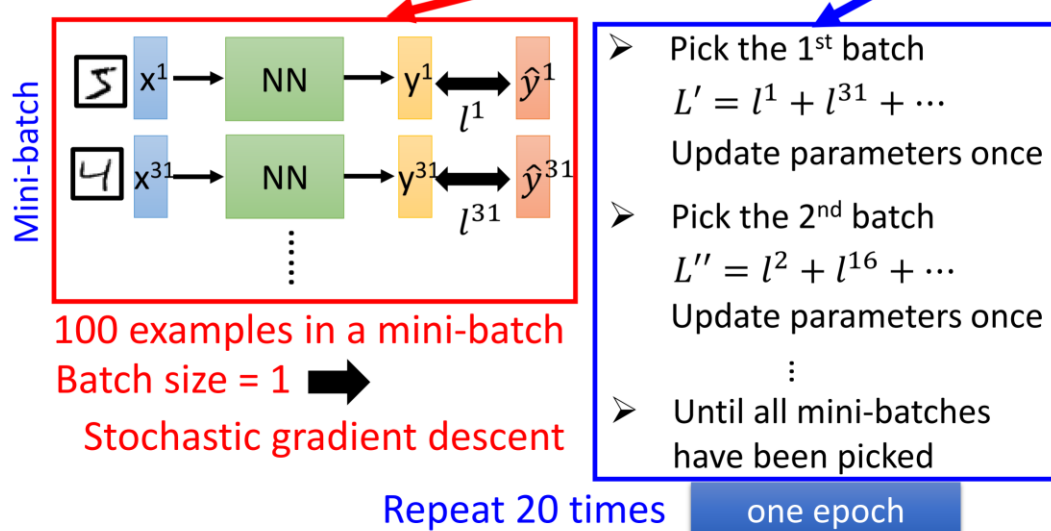# Mini-batch

Batch size influences both *speed* and *performance*. You have to tune it.

```
model.fit(x_train, y_train, batch_size=100, nb_epoch=20)
```


Mini-batch

$x^1 \rightarrow$ NN $\rightarrow y^1 \longleftrightarrow \hat{y}^1$
l^1

$x^{31} \rightarrow$ NN $\rightarrow y^{31} \longleftrightarrow \hat{y}^{31}$
l^{31}

100 examples in a mini-batch

Batch size = 1 ➡

Stochastic gradient descent

➢ Pick the 1st batch
$$L' = l^1 + l^{31} + \cdots$$
Update parameters once

➢ Pick the 2nd batch
$$L'' = l^2 + l^{16} + \cdots$$
Update parameters once

➢ Until all mini-batches have been picked

Repeat 20 times — one epoch

Speed - Matrix Operation

• Why mini-batch is faster than stochastic gradient descent?

Stochastic Gradient Descent

$z^1 = W^1 x$ $z^1 = W^1 x$ ......

Mini-batch

matrix

$z^1 \; z^1 = W^1 \; x \; x$

Practically, which one is faster?