Pytorch-like API

```
D_in, H, D_out = 2, 3, 2

mlp = MLP([
    Linear(D_in, H),
    ReLU(),
    Linear(H, D_out)
])

optimizer = SGD(mlp, lr=0.1)
loss = CrossEntropy(mlp)

epochs = 100
for e in range(epochs):
  for x, y in zip(X, Y):
    y_pred = mlp(x)
    loss(y_pred, y)
    loss.backward()
    optimizer.update()
```

MLP

```
class MLP:
    def __init__(self, layers):
        self.layers = layers

    def __call__(self, x):
        for layer in self.layers:
            x = layer(x)
        return x
```