Losses

```python
class CrossEntropy():
    def __init__(self, net):
        self.net = net

    def __call__(self, output, target):
        self.output, self.target = output, target
        logits = output[np.arange(len(output)), target]
        loss = - logits + np.log(np.sum(np.exp(output), axis=-1))
        loss = loss.mean()
        return loss

    def grad_crossentropy(self):
        answers = np.zeros_like(self.output)
        answers[np.arange(len(self.output)), self.target] = 1
        return (- answers + softmax(self.output)) / self.output.shape[0]

    def backward(self):
        grad = self.grad_crossentropy()
        for layer in reversed(self.net.layers):
            grad = layer.backward(grad)
```