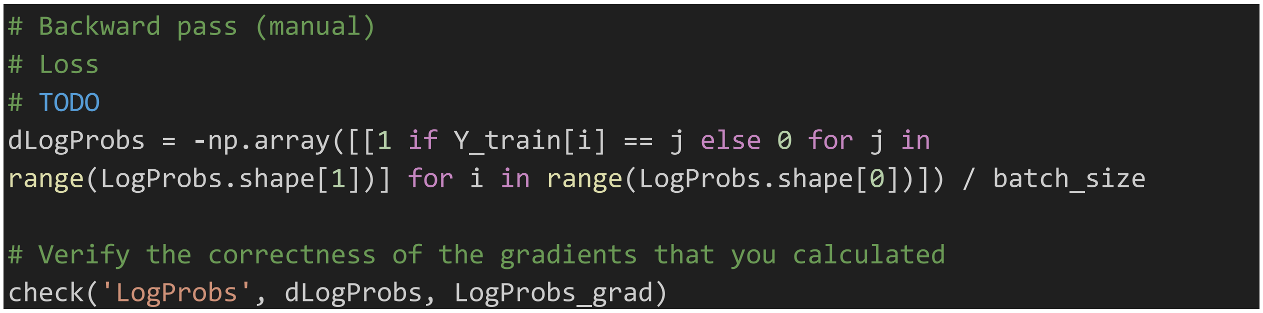


```
# Backward pass (manual)
# Loss
# TODO
dLogProbs = -np.array([[1 if Y_train[i] == j else 0 for j in
range(LogProbs.shape[1])] for i in range(LogProbs.shape[0])]) / batch_size

# Verify the correctness of the gradients that you calculated
check('LogProbs', dLogProbs, LogProbs_grad)
```

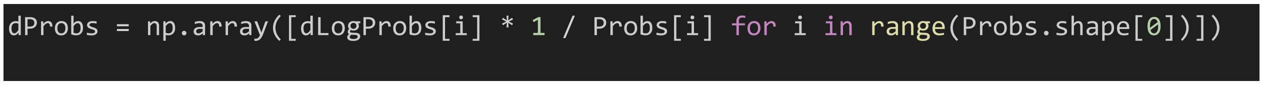
$$m = batch\ size$$

خروجی مدل $= \ 32 * 10$

$$L = -\frac{1}{m}\sum logp_{i,y_i} \quad y_i = ground\ truth\ index$$

$$L = -\frac{1}{m}\sum_{i}^{m}\sum_{j} y_j \log_{p_{i,j}}$$

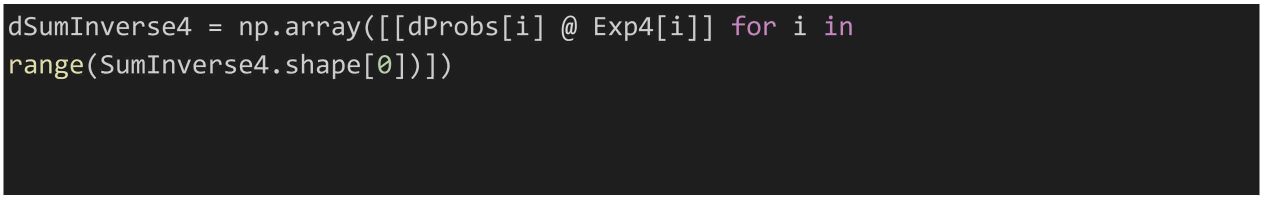
$$dLogProbs = \frac{\partial L}{\partial LogProbs_{i,j}} = -\frac{1}{m}\begin{cases}1 & j = y_i \\ 0 & j \neq y_i\end{cases}$$



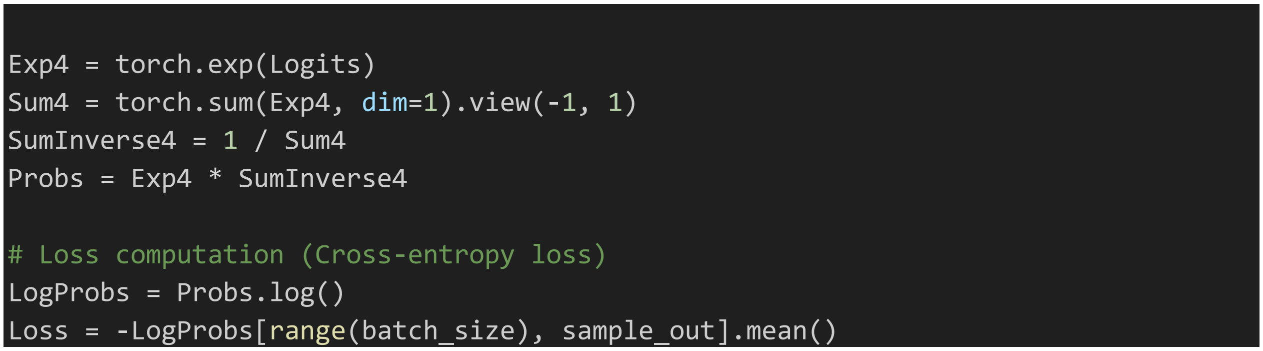
```
dProbs = np.array([dLogProbs[i] * 1 / Probs[i] for i in range(Probs.shape[0])])
```

$$dProbs = \frac{\partial L}{\partial Probs_{i,j}} = \frac{\partial L}{\partial LogProbs_{i,j}} \times \frac{\partial LogProbs_{i,j}}{\partial Probs_{i,j}}$$

$$= \ dLogProbs_{i,j} \times \frac{1}{Probs_{i,j}}$$



```
dSumInverse4 = np.array([[dProbs[i] @ Exp4[i]] for i in
range(SumInverse4.shape[0])])
```

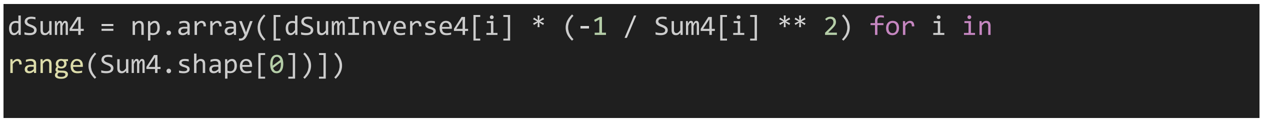


```
Exp4 = torch.exp(Logits)
Sum4 = torch.sum(Exp4, dim=1).view(-1, 1)
SumInverse4 = 1 / Sum4
Probs = Exp4 * SumInverse4

# Loss computation (Cross-entropy loss)
LogProbs = Probs.log()
Loss = -LogProbs[range(batch_size), sample_out].mean()
```

$$Probs_{i,j} = SumInverse4_i * \; Exp4_{i,j}$$

$$\frac{\partial L}{\partial SumInverse4_i} = \sum_j \frac{\partial L}{\partial Probs_{i,j}} * \; Exp4_{i,j}$$

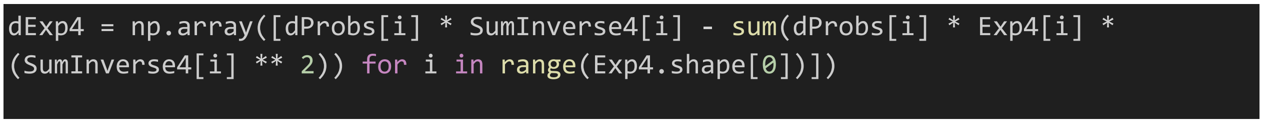


```
dSum4 = np.array([dSumInverse4[i] * (-1 / Sum4[i] ** 2) for i in
range(Sum4.shape[0])])
```

$$Sum4_i = \frac{1}{SumInverse4_i}$$

$$\frac{\partial L}{\partial Sum4_i} = \frac{\partial L}{\partial SumInverse4_i} \times \frac{\partial SumInverse4_i}{\partial Sum4_i}$$

$$= dSumInvers4_i \times \left(-\frac{1}{(Sum4_i)^2} \right)$$

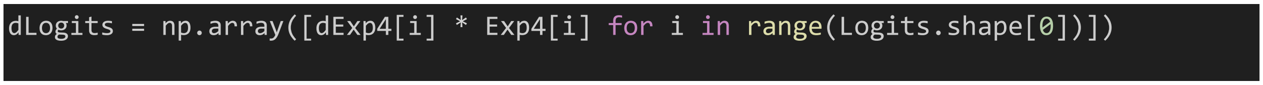


```
dExp4 = np.array([dProbs[i] * SumInverse4[i] - sum(dProbs[i] * Exp4[i] *
(SumInverse4[i] ** 2)) for i in range(Exp4.shape[0])])
```

$$Probs_{i,j} = SumInverse4_i * \; Exp4_{i,j}$$

$$Sum4_i = \sum_j Exp4_{i,j}$$

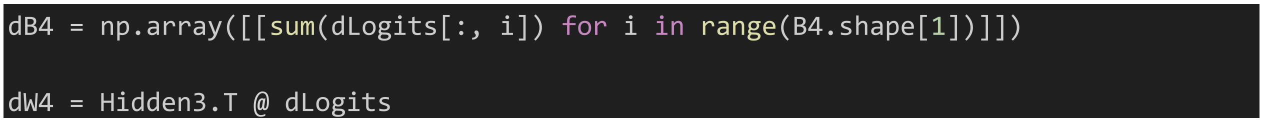
$$\begin{aligned}\frac{\partial L}{\partial Exp4_{i,j}} &= \frac{\partial L}{\partial Prob_{i,j}} \times \frac{\partial Prob_{i,j}}{\partial Exp4_i} \\ &+ \frac{\partial L}{\partial Prob_{i,j}} \times \frac{\partial Prob4_i}{\partial SumInverse4_i} \times \frac{\partial SumInverse4_i}{\partial Sum4_i} \\ &\times \frac{\partial Sum4_i}{\partial Exp4_{i,j}} \\ &= dProbs_{i,j} \times SumInverse4_i + dProbs_{i,j} \\ &\qquad \times \sum_j Exp4_{i,j} \times \left(-\frac{1}{(Sum4_i)^2}\right) \times 1\end{aligned}$$



```
dLogits = np.array([dExp4[i] * Exp4[i] for i in range(Logits.shape[0])])
```

$$Exp4_{i,j} = e^{Logit_{i,j}}$$

$$\begin{aligned}\frac{\partial L}{\partial Logit_{i,j}} &= \frac{\partial L}{\partial Exp4_{i,j}} \times \frac{\partial Exp4_{i,j}}{\partial Logit_{i,j}} \\ &= dExp4_{i,j} \times Exp4_{i,j}\end{aligned}$$



```
dB4 = np.array([[sum(dLogits[:, i]) for i in range(B4.shape[1])]])

dW4 = Hidden3.T @ dLogits
```

$$Logit = H^{(3)} @ W^{(4)} + B^{(4)}$$

$$Logit_{i,j} = \sum_{k=1}^{n_3} H_{i,k}^{(3)} W_{k,j}^{(4)} + B_j^{(4)}$$

$$\frac{\partial L}{\partial B_j^{(4)}} = \sum_{i=1}^{m} \frac{\partial L}{\partial Logit_{i,j}^{(4)}}$$

$$\frac{\partial L}{\partial W^{(4)}} = {H^{(3)}}^{T} \mathbin{@} dLogit$$