BatchNorm

Juan David Martinez Vargas

August 2024

1 Introduction

In this document, we provide a detailed explanation of the `BatchNorm1d` and `BatchNorm2d` functions in PyTorch. We describe their purposes, how they work, and the mathematical operations they perform.

2 BatchNorm1d

2.1 Input Tensor Dimensions

The input tensor has dimensions (N, C) or (N, C, L), where:

- N is the batch size.
- C is the number of channels (or features).
- L is the length of the sequence (in the case of a 3D input tensor).

2.2 Per-Channel Normalization

For each channel c (where c ranges from 1 to C), `BatchNorm1d` computes the mean μ_c and variance σ_c^2 over the entire batch and sequence length (if applicable). These statistics are computed as follows:

For a 2D input tensor (N, C):

$$\mu_c = \frac{1}{N} \sum_{n=1}^{N} x_{n,c}$$

$$\sigma_c^2 = \frac{1}{N} \sum_{n=1}^{N} (x_{n,c} - \mu_c)^2$$

For a 3D input tensor (N, C, L):

$$\mu_c = \frac{1}{N \times L} \sum_{n=1}^{N} \sum_{l=1}^{L} x_{n,c,l}$$

$$\sigma_c^2 = \frac{1}{N \times L} \sum_{n=1}^{N} \sum_{l=1}^{L} (x_{n,c,l} - \mu_c)^2$$

These statistics are then used to normalize each element in the channel.

2.3 Normalization

The normalization step for each element $x_{n,c}$ (or $x_{n,c,l}$ in the case of a 3D tensor) is given by:

$$\hat{x}_{n,c} = \frac{x_{n,c} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

where ϵ is a small constant added for numerical stability.

2.4 Learnable Parameters

After normalization, each channel has learnable parameters γ_c (scale) and β_c (shift), both of size C. The final output is obtained by applying these parameters:

$$y_{n,c} = \gamma_c \hat{x}_{n,c} + \beta_c$$

(or $y_{n,c,l} = \gamma_c \hat{x}_{n,c,l} + \beta_c$ for 3D input tensors).

2.5 Summary for BatchNorm1d

- **Mean and Variance:** `BatchNorm1d` computes a mean and variance for each channel across the entire batch (and sequence length if applicable).
- **Normalization:** Each element within a channel is normalized using the channel's mean and variance.
- **Scaling and Shifting:** The normalized output is then scaled and shifted by learnable parameters γ_c and β_c, which are specific to each channel.

3 BatchNorm2d

3.1 Input Tensor Dimensions

The input tensor has dimensions (N, C, H, W), where:

- N is the batch size.
- C is the number of channels.
- H and W are the height and width of each feature map.

3.2 Per-Channel Normalization

For each channel c (where c ranges from 1 to C), `BatchNorm2d` computes the mean μ_c and variance σ_c^2 over the entire batch and spatial dimensions. These statistics are computed as follows:

$$\mu_c = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} x_{n,c,h,w}$$

$$\sigma_c^2 = \frac{1}{N \times H \times W} \sum_{n=1}^{N} \sum_{h=1}^{H} \sum_{w=1}^{W} (x_{n,c,h,w} - \mu_c)^2$$

These statistics are then used to normalize each element in the channel.

3.3 Normalization

The normalization step for each element $x_{n,c,h,w}$ is given by:

$$\hat{x}_{n,c,h,w} = \frac{x_{n,c,h,w} - \mu_c}{\sqrt{\sigma_c^2 + \epsilon}}$$

where ϵ is a small constant added for numerical stability.

3.4 Learnable Parameters

After normalization, each channel has learnable parameters γ_c (scale) and β_c (shift), both of size C. The final output is obtained by applying these parameters:

$$y_{n,c,h,w} = \gamma_c \hat{x}_{n,c,h,w} + \beta_c$$

This allows the model to learn to scale and shift the normalized output, providing flexibility.

3.5 Summary for BatchNorm2d

- **Mean and Variance:** `BatchNorm2d` computes a mean and variance for each channel across the entire batch and spatial dimensions.
- **Normalization:** Each element within a channel is normalized using the channel's mean and variance.
- **Scaling and Shifting:** The normalized output is then scaled and shifted by learnable parameters γ_c and β_c, which are specific to each channel.