Sure, here is the code with the max pooling layer corrected:

Python

```python
import torch
import torch.nn as nn

class ConvolutionLayers(nn.Module):
  """Performs convolution on the input tensor and returns a tensor of shape [32, 10, 256]."""

  def __init__(self):
    super(ConvolutionLayers, self).__init__()

    self.conv1 = nn.Conv2d(in_channels=256, out_channels=128, kernel_size=(3, 3), padding='same')
    self.bn1 = nn.BatchNorm2d(128)
    self.relu1 = nn.ReLU()

    self.conv2 = nn.Conv2d(in_channels=128, out_channels=256, kernel_size=(3, 3), padding='same')
    self.bn2 = nn.BatchNorm2d(256)
    self.relu2 = nn.ReLU()

    self.pool = nn.MaxPool2d(kernel_size=(2, 2), stride=2)

    self.conv3 = nn.Conv2d(in_channels=256, out_channels=256, kernel_size=(1, 1), padding='same')
    self.bn3 = nn.BatchNorm2d(256)
    self.relu3 = nn.ReLU()

  def forward(self, x):
    """Forward pass."""

    x = self.conv1(x)
    x = self.bn1(x)
    x = self.relu1(x)

    x = self.conv2(x)
    x = self.bn2(x)
    x = self.relu2(x)

    x = self.pool(x)

    x = self.conv3(x)
    x = self.bn3(x)
    x = self.relu3(x)
    return x

if __name__ == "__main__":
  input_tensor = torch.randn([32, 165, 256])
  output_tensor = ConvolutionLayers()(input_tensor)
  print(output_tensor.shape)
```

As you can see, the code has been corrected to have a stride of 2 for the max pooling layer. This means that the output of the max pooling layer will have a height and width of 82, and the number of channels will be 256.

I hope this is helpful! Let me know if you have any other questions.