MONASH University

FIT3181/5215 Deep Learning

Quiz for:
Advanced Convolutional Neural Networks

Trung Le and Teaching team

Department of Data Science and AI
Faculty of Information Technology, Monash University
Email: **trunglm@monash.edu**

GROUP OF EIGHT AUSTRALIA

Question 1

Which statements are correct? (MC)

- A. In traditional approach, the training signal from classifier can be used to improve feature extractor.
- B. In deep learning approach, the training signal from classifier can be used to improve feature extractor.
- C. In traditional approach, the training signal from classifier cannot be used to improve feature extractor.
- D. In deep learning approach, the training signal from classifier cannot be used to improve feature extractor.

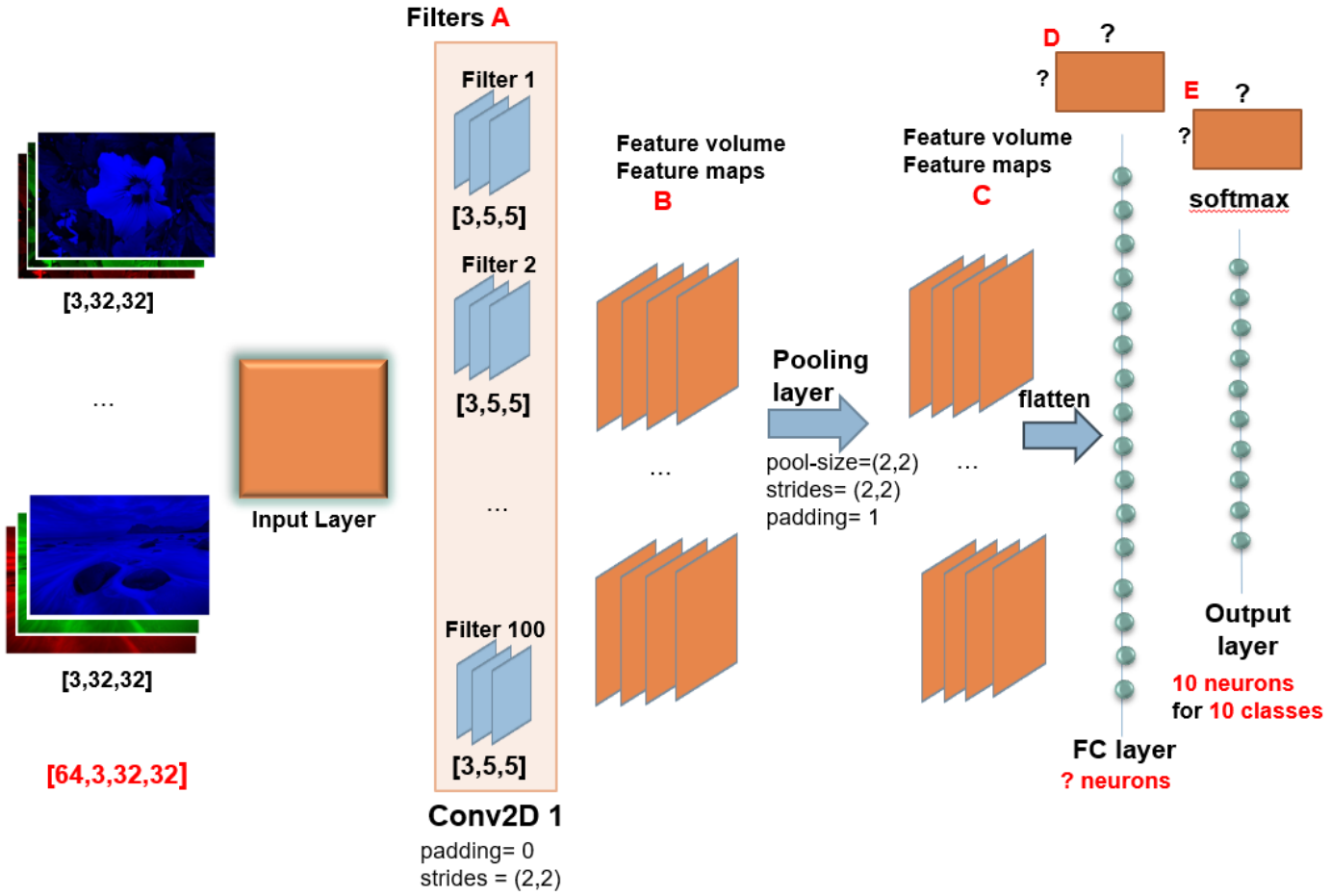
Question 1

Which statements are correct? (MC)

- A. In traditional approach, the training signal from classifier can be used to improve feature extractor.
- B. In deep learning approach, the training signal from classifier can be used to improve feature extractor. **[x]**
- C. In traditional approach, the training signal from classifier cannot be used to improve feature extractor. **[x]**
- D. In deep learning approach, the training signal from classifier cannot be used to improve feature extractor.

Question 2

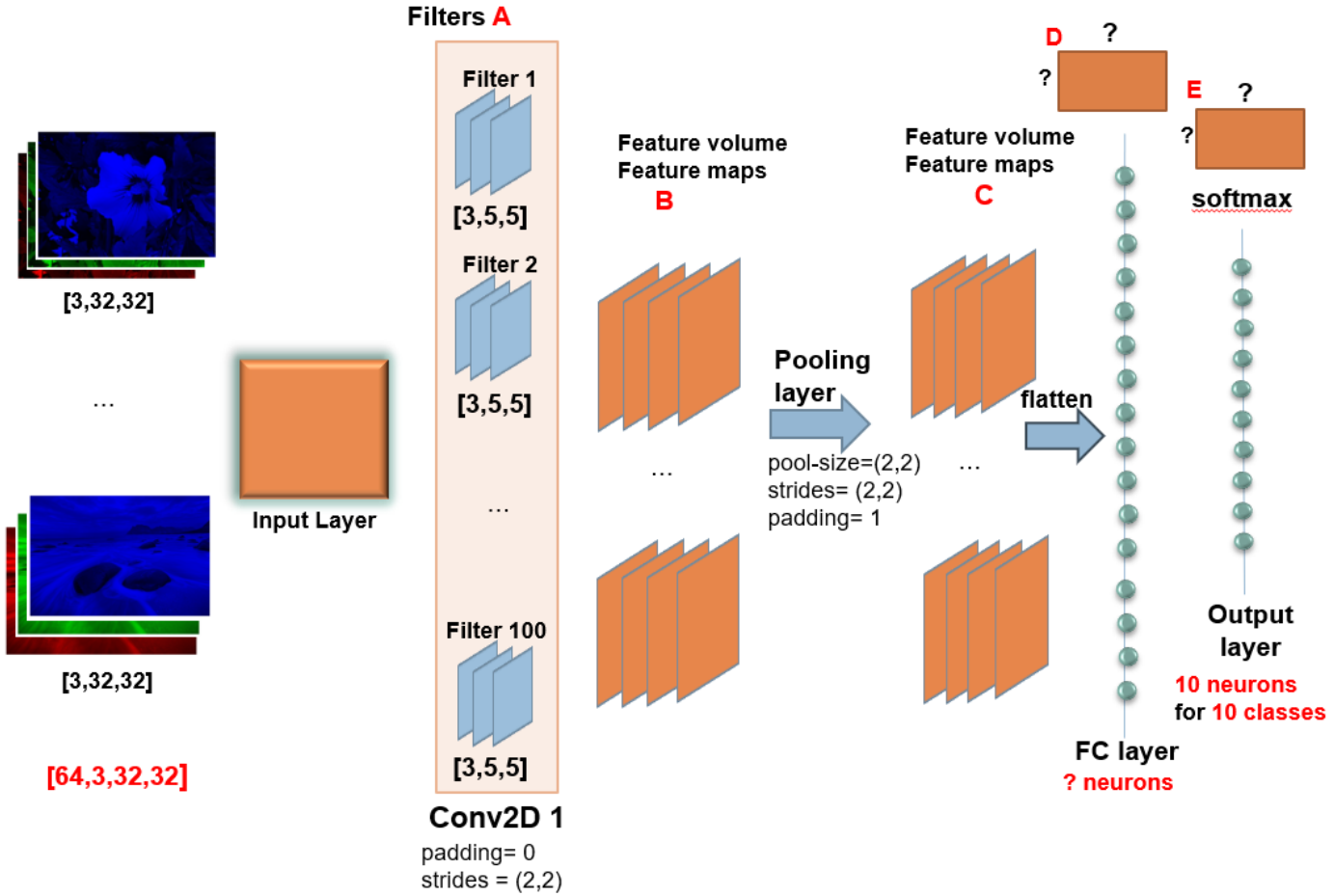
What are the shapes of tensors in A, B, C, D and E?



- A. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100x8x8], [64, 10]
- B. [100,3,5,5], [64,100,14,14], [64,100,7,7], [1, 64x100x7x7], [64, 10]
- C. [64,3,5,5], [64,100,14,14], [64,100,7,7], [64, 100x7x7], [64, 10]
- D. [100,3,5,5], [64,100,15,15], [64,100,7,7], [64, 100x7x7], [64, 10]

Question 2

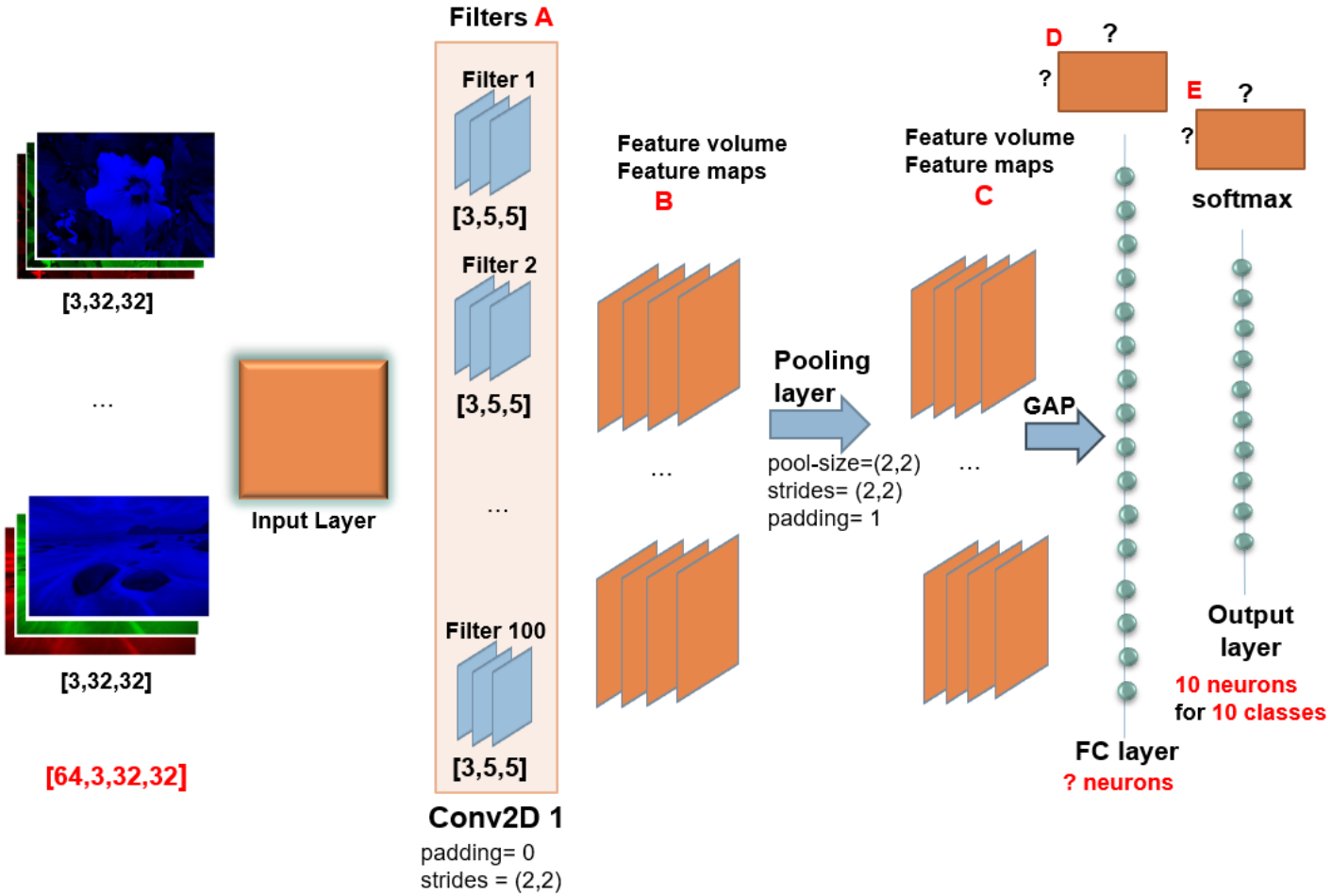
What are the shapes of tensors in A, B, C, D and E?



- A. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100x8x8], [64, 10] **[x]**
- B. [100,3,5,5], [64,100,14,14], [64,100,7,7], [1, 64x100x7x7], [64, 10]
- C. [64,3,5,5], [64,100,14,14], [64,100,7,7], [64, 100x7x7], [64, 10]
- D. [100,3,5,5], [64,100,15,15], [64,100,7,7], [64, 100x7x7], [64, 10]

Question 3

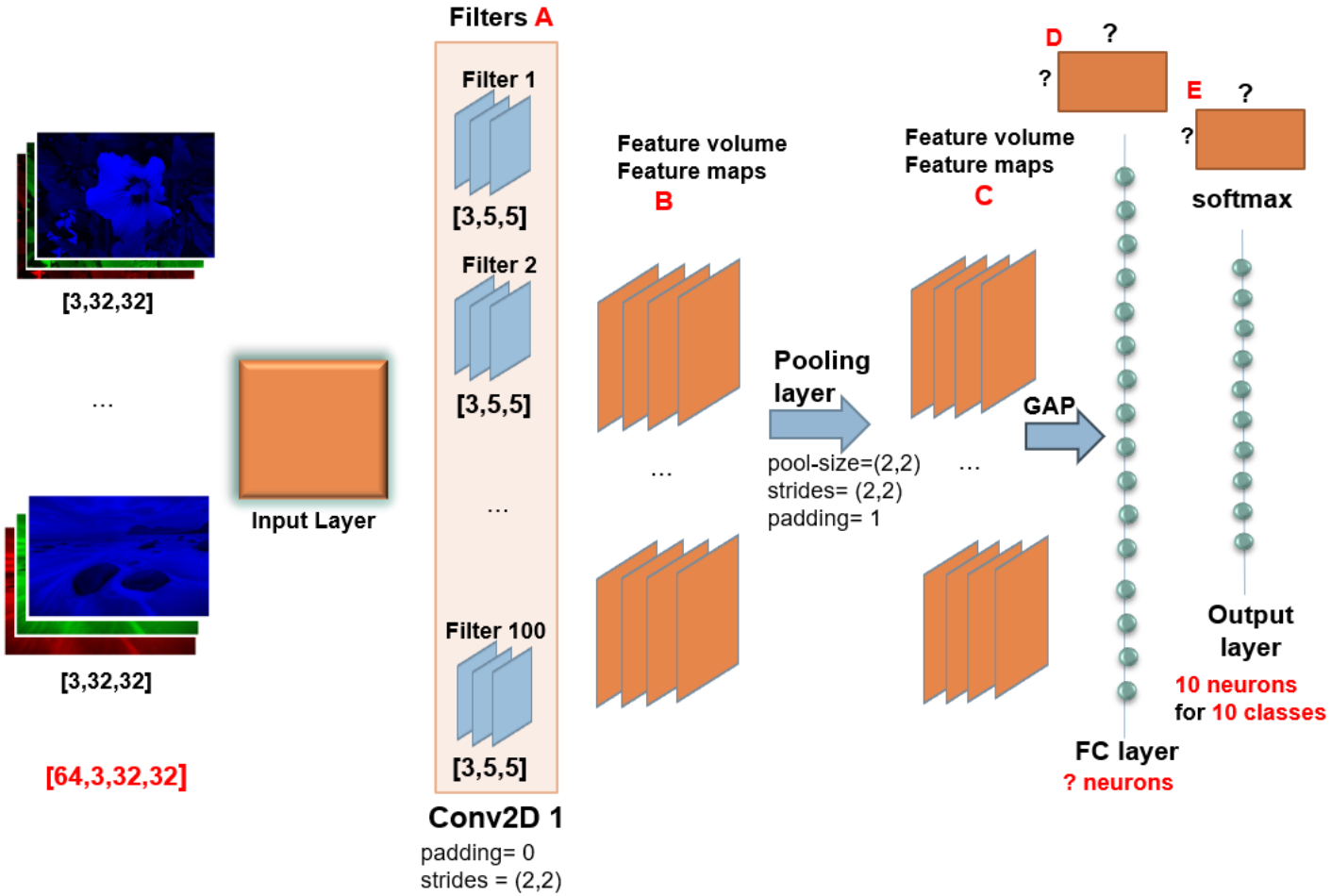
What are the shapes of tensors in A, B, C, D and E?



- A. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100x8x8], [64, 10]
- B. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100], [64, 10]
- C. [64,3,5,5], [64,100,14,14], [64,100,7,7], [64, 100x7x7], [64, 10]
- D. [100,3,5,5], [64,100,15,15], [64,100,7,7], [64, 100x7x7], [64, 10]

Question 3

What are the shapes of tensors in A, B, C, D and E?



- A. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100x8x8], [64, 10]
- B. [100,3,5,5], [64,100,14,14], [64,100,8,8], [64, 100], [64, 10] **[x]**
- C. [64,3,5,5], [64,100,14,14], [64,100,7,7], [64, 100x7x7], [64, 10]
- D. [100,3,5,5], [64,100,15,15], [64,100,7,7], [64, 100x7x7], [64, 10]

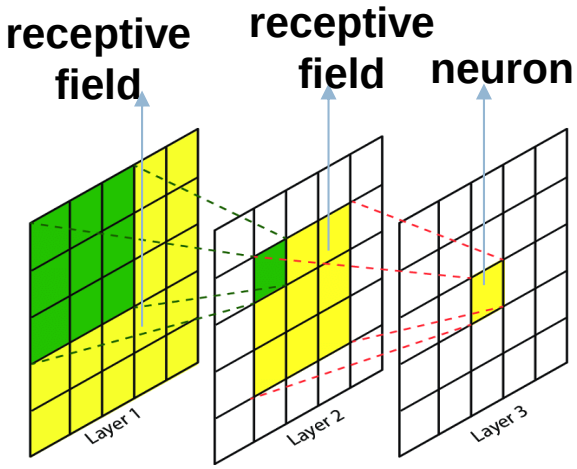
Question 4

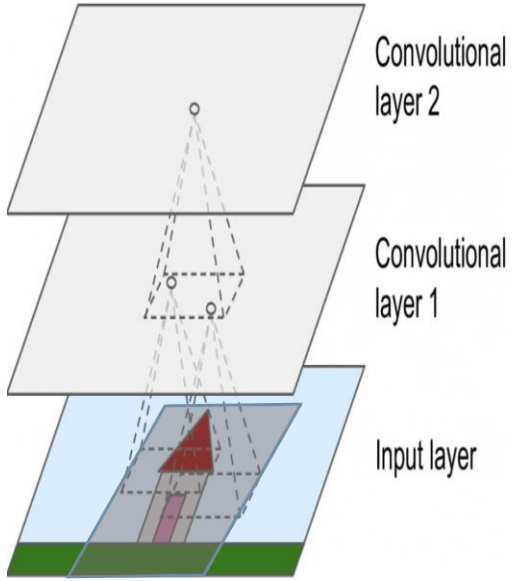
What are correct statements about the receptive field? (MC)

- A. Receptive field of neurons on higher layers become smaller.
- B. The value of a neuron is not computationally relevant to its receptive field.
- C. Receptive field of neurons on higher layers become larger.
- D. The value of a neuron is computationally relevant to its receptive field.

Question 4

What are correct statements about the receptive field? (MC)

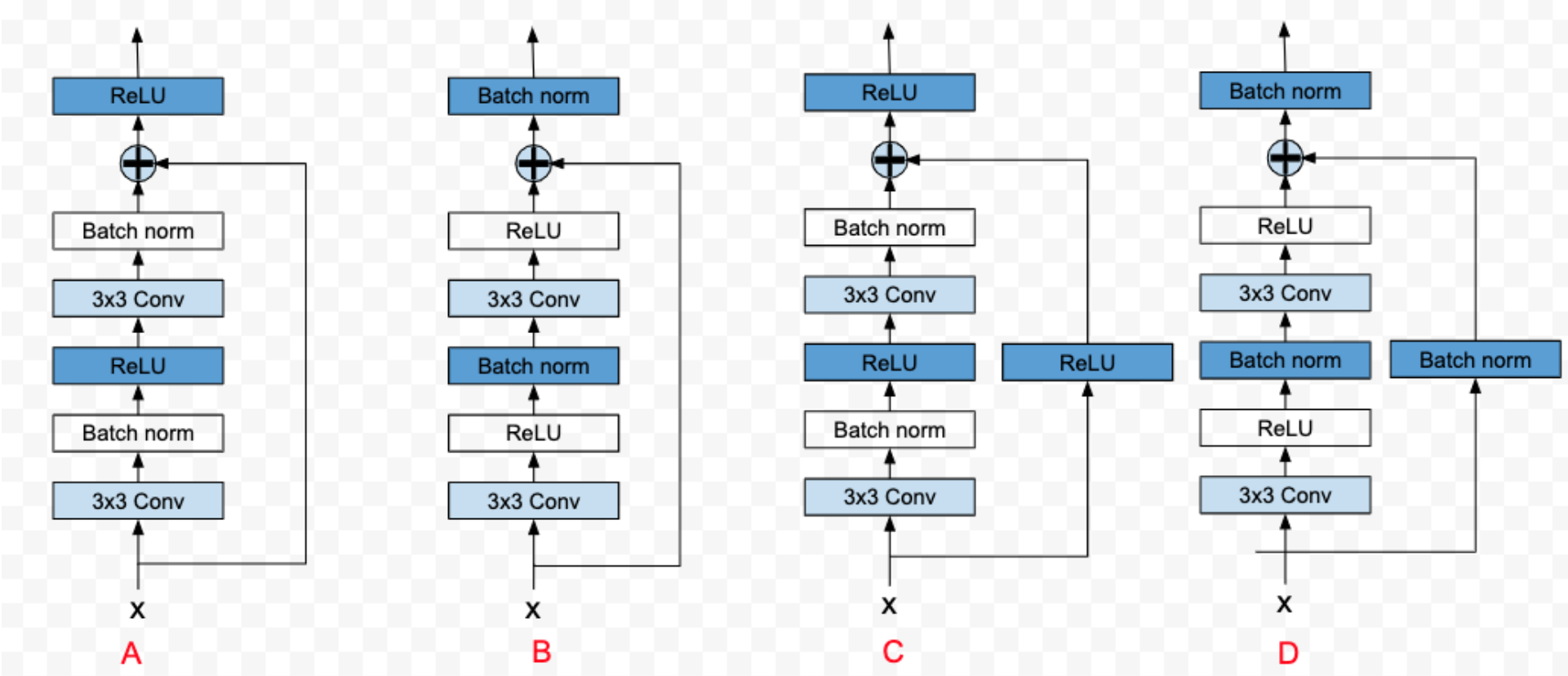




- A. Receptive field of neurons on higher layers become smaller.
- B. The value of a neuron is not computationally relevant to its receptive field.
- C. Receptive field of neurons on higher layers become larger. **[x]**
- D. The value of a neuron is computationally relevant to its receptive field. **[x]**

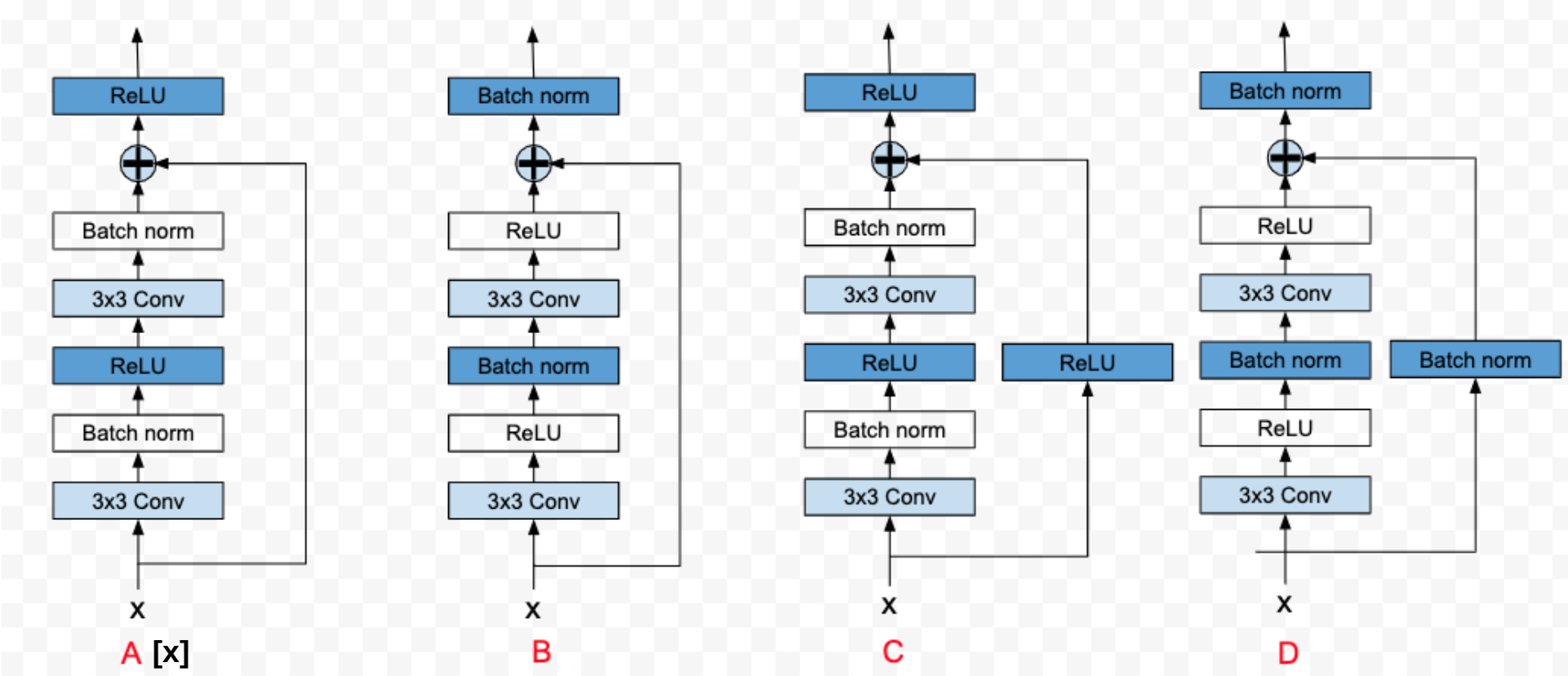
Question 5

Which illustration is correct for the residual block? (SC).



Question 5

Which illustration is correct for the residual block? (SC).



Question 6

Given an implementation of the residual block as below? What is the shape of Y (SC).

A. [10,3, 32,32]
B. [10,3, 16,16]
C. [3,3,32,32]
D. Raise an error.

```
class Residual(nn.Module):
    def __init__(self, num_channels, use_1x1conv=False, strides=1):
        super().__init__()
        self.conv1 = nn.LazyConv2d(num_channels, kernel_size=3, stride=strides, padding=1)
        self.conv2 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1)
        self.conv3 = None
        if use_1x1conv:
            self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1, stride=strides)
        self.bn1 = nn.BatchNorm2d(num_channels)
        self.bn2 = nn.BatchNorm2d(num_channels)
        self.relu = nn.ReLU()

    def forward(self, X):
      Y = self.relu(self.bn1(self.conv1(X)))
      Y = self.bn2(self.conv2(Y))
      if self.conv3 is not None:
        X = self.conv3(X)
      Y += X
      return self.relu(Y)
```

```
blk = Residual(num_channels=3, use_1x1conv=False, strides=1)
X = torch.rand((10, 3, 32, 32))
Y = blk(X)
print(Y.shape)
```

Question 6

Given an implementation of the residual block as below? What is the shape of Y (SC).

A. [10,3,32,32] **[x]**
B. [10,3, 16,16]
C. [3, 3, 32,32]
D. Raise an error.

```
class Residual(nn.Module):
    def __init__(self, num_channels, use_1x1conv=False, strides=1):
        super().__init__()
        self.conv1 = nn.LazyConv2d(num_channels, kernel_size=3, stride=strides, padding=1)
        self.conv2 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1)
        self.conv3 = None
        if use_1x1conv:
            self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1, stride=strides)
        self.bn1 = nn.BatchNorm2d(num_channels)
        self.bn2 = nn.BatchNorm2d(num_channels)
        self.relu = nn.ReLU()

    def forward(self, X):
      Y = self.relu(self.bn1(self.conv1(X)))
      Y = self.bn2(self.conv2(Y))
      if self.conv3 is not None:
        X = self.conv3(X)
      Y += X
      return self.relu(Y)
```

```
blk = Residual(num_channels=3, use_1x1conv=False, strides=1)
X = torch.rand((10, 3, 32, 32))
Y = blk(X)
print(Y.shape)
```

Question 7

Given an implementation of the residual block as below? What is the shape of Y (SC).

A. [10,3, 32,32]
B. [10,6, 16,16]
C. [3,3,32,32]
D. Raise an error.

```
class Residual(nn.Module):
    def __init__(self, num_channels, use_1x1conv=False, strides=1):
        super().__init__()
        self.conv1 = nn.LazyConv2d(num_channels, kernel_size=3, stride=strides, padding=1)
        self.conv2 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1)
        self.conv3 = None
        if use_1x1conv:
            self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1, stride=strides)
        self.bn1 = nn.BatchNorm2d(num_channels)
        self.bn2 = nn.BatchNorm2d(num_channels)
        self.relu = nn.ReLU()

    def forward(self, X):
      Y = self.relu(self.bn1(self.conv1(X)))
      Y = self.bn2(self.conv2(Y))
      if self.conv3 is not None:
        X = self.conv3(X)
      Y += X
      return self.relu(Y)
```

```
blk = Residual(6, use_1x1conv=True, strides=2)
X = torch.rand((10, 3, 32, 32))
Y = blk(X)
print(Y.shape)
```

Question 7

Given an implementation of the residual block as below? What is the shape of Y (SC).

A. [10,3,32,32]
B. [10,6,16,16] **[x]**
C. [3,3,32,32]
D. Raise an error.

```
class Residual(nn.Module):
    def __init__(self, num_channels, use_1x1conv=False, strides=1):
        super().__init__()
        self.conv1 = nn.LazyConv2d(num_channels, kernel_size=3, stride=strides, padding=1)
        self.conv2 = nn.LazyConv2d(num_channels, kernel_size=3, padding=1)
        self.conv3 = None
        if use_1x1conv:
            self.conv3 = nn.LazyConv2d(num_channels, kernel_size=1, stride=strides)
        self.bn1 = nn.BatchNorm2d(num_channels)
        self.bn2 = nn.BatchNorm2d(num_channels)
        self.relu = nn.ReLU()

    def forward(self, X):
      Y = self.relu(self.bn1(self.conv1(X)))
      Y = self.bn2(self.conv2(Y))
      if self.conv3 is not None:
        X = self.conv3(X)
      Y += X
      return self.relu(Y)
```

```
blk = Residual(6, use_1x1conv=True, strides=2)
X = torch.rand((10, 3, 32, 32))
Y = blk(X)
print(Y.shape)
```

Question 8

*Given an implementation of the ResNet as below. Assume that we are feeding a batch **[32, 3, 64, 64]** to our ResNet. What are the shape of A,B,C,D, E, and F? (SC).*

A. [32,64,16,16], [32,64,16,16], [32,128,8,8], [32,256,4,4], [32,256,1,1], [32,10]

B. [32,64,16,16], [32,64,16,16], [32,128,8,8], [32,256,4,4], [32,256], [32,10]

C. [32,64,16,16], [32,64,8,8], [32,128,4,4], [32,256,2,2], [32,256,1,1], [32,10]

D. Raise an error.

```
class create_ResNet(nn.Module):
  def __init__(self):
    super().__init__()
    self.layers = nn.ModuleList([
                  nn.LazyConv2d(64, kernel_size=7, stride=2, padding=3),
                  nn.LazyBatchNorm2d(),
                  nn.ReLU(),
          A <-    nn.MaxPool2d(kernel_size=3, stride=2, padding=1),
          B <-    ResnetBlock(64, 2, first_block=True),
          C <-    ResnetBlock(128, 2),
          D <-    ResnetBlock(256, 2),
          E <-    nn.AdaptiveAvgPool2d((1, 1)),
                  nn.Flatten(1),
          F <-    nn.LazyLinear(10),
                  # nn.Softmax(dim=-1)
                  ])
  def forward(self, X):
    for _, layer in enumerate(self.layers):
      X = layer(X)
    return X
```

Question 8

*Given an implementation of the ResNet as below. Assume that we are feeding a batch **[32, 3, 64, 64]** to our ResNet. What are the shape of A,B,C,D, E, and F? (SC).*

A. [32,64,16,16], [32,64,16,16], [32,128,8,8], [32,256,4,4], [32,256,1,1], [32,10] **[x]**

B. [32,64,16,16], [32,64,16,16], [32,128,8,8], [32,256,4,4], [32,256], [32,10]

C. [32,64,16,16], [32,64,8,8], [32,128,4,4], [32,256,2,2], [32,256,1,1], [32,10]

D. Raise an error.

```
class create_ResNet(nn.Module):
  def __init__(self):
    super().__init__()
    self.layers = nn.ModuleList([
                  nn.LazyConv2d(64, kernel_size=7, stride=2, padding=3),
                  nn.LazyBatchNorm2d(),
                  nn.ReLU(),
          A <-    nn.MaxPool2d(kernel_size=3, stride=2, padding=1),
          B <-    ResnetBlock(64, 2, first_block=True),
          C <-    ResnetBlock(128, 2),
          D <-    ResnetBlock(256, 2),
          E <-    nn.AdaptiveAvgPool2d((1, 1)),
                  nn.Flatten(1),
          F <-    nn.LazyLinear(10),
                  # nn.Softmax(dim=-1)
                  ])
  def forward(self, X):
    for _, layer in enumerate(self.layers):
      X = layer(X)
    return X
```

Question 9

Which statements are correct for ResNet architecture? (MC).

- A. In ResNet architecture, ReLU activation function is followed by Batch Normalization layer.
- B. It is possible to replace ReLU by Sigmoid activation function because of the skip-connection can help to reduce gradient vanishing.
- C. 1x1 Conv in skip-connection is used to change number of output channels.
- D. A ResNet model consists of many ResNet blocks, each ResNet block consists of many residual blocks, each residual block includes several convolutional and activation layers.

Question 9

Which statements are correct for ResNet architecture? (MC).

- A. In ResNet architecture, ReLU activation function is followed by Batch Normalization layer.
- B. It is possible to replace ReLU by Sigmoid activation function because of the skip-connection can help to reduce gradient vanishing.
- C. 1x1 Conv in skip-connection is used to change number of output channels. **[x]**
- D. A ResNet model consists of many ResNet blocks, each ResNet block consists of many residual blocks, each residual block includes several convolutional and activation layers. **[x]**

Question 10

Given an adversarial example x_{adv} of a clean example x w.r.t. model f, $y \in \{1,2,\ldots,M\}$ is the true label. Which statements are correct? (MC).

- A. x_{adv} and x look very similar under human perspective
- B. x_{adv} and x look very different under human perspective
- C. $argmax_{1\leq m\leq M} f_m(x_{adv}) = argmax_{1\leq m\leq M} f_m(x)$
- D. $argmax_{1\leq m\leq M} f_m(x_{adv}) \neq argmax_{1\leq m\leq M} f_m(x)$

Question 10

Given an adversarial example x_{adv} of a clean example x w.r.t. model f, $y \in \{1,2,\ldots,M\}$ is the true label. Which statements are correct? (MC).

- A. x_{adv} and x look very similar under human perspective **[x]**
- B. x_{adv} and x look very different under human perspective
- C. $argmax_{1\leq m\leq M} f_m(x_{adv}) = argmax_{1\leq m\leq M} f_m(x)$
- D. $argmax_{1\leq m\leq M} f_m(x_{adv}) \neq argmax_{1\leq m\leq M} f_m(x)$ **[x]**

Question 11

Given a constraint of an adversarial example as follow: $x_{adv} \in B_\epsilon(x) = \{x' : \|x' - x\|_\infty \leq \epsilon\}$. *Which statements are correct? (MC)*

- A. This constraint to make sure that x_{adv} and x look very similar under human perspective
- B. This constraint to make sure that x_{adv} and x look very different under human perspective
- C. This constraint to make sure that $argmax_{1 \leq m \leq M} f_m(x_{adv}) = argmax_{1 \leq m \leq M} f_m(x)$
- D. The highest absolute difference between pixels of x_{adv} and x is less than or equal ϵ

Question 11

Given a constraint of an adversarial example as follow: $x_{adv} \in B_\epsilon(x) = \{x' : \|x' - x\|_\infty \leq \epsilon\}$. *Which statements are correct? (MC)*

- A. This constraint to make sure that x_{adv} and x look very similar under human perspective **[x]**
- B. This constraint to make sure that x_{adv} and x look very different under human perspective
- C. This constraint to make sure that $argmax_{1 \leq m \leq M} f_m(x_{adv}) = argmax_{1 \leq m \leq M} f_m(x)$
- D. The highest absolute difference between pixels of x_{adv} and x is less than or equal ϵ **[x]**

Question 12

Given a DL model $f(x;\theta)$ parameterized by θ where $f(x;\theta)$ represents the prediction probabilities of x associated with a ground-truth label $y \in \{1, \ldots, M\}$, we find an adversarial example by $\boldsymbol{x_{adv} = argmax_{x' \in B_\epsilon(x)} l(f(x';\theta), y)}$. Which statements are correct? (MC)

- A. We maximally increase the chance to predict x_{adv} with label y.
- B. We maximally decrease the chance to predict x_{adv} with label y.
- C. We maximally increase the chance to predict x_{adv} with any else label $y' \neq y$.
- D. It is a targeted attack.
- E. It is an untargeted attack.

Question 12

Given a DL model $f(x;\theta)$ parameterized by θ where $f(x;\theta)$ represents the prediction probabilities of x associated with a ground-truth label $y \in \{1, \ldots, M\}$, we find an adversarial example by $\boldsymbol{x_{adv} = argmax_{x' \in B_\epsilon(x)} l(f(x';\theta), y)}$. Which statements are correct? (MC)

- A. We maximally increase the chance to predict x_{adv} with label y.
- B. We maximally decrease the chance to predict x_{adv} with label y. **[x]**
- C. We maximally increase the chance to predict x_{adv} with any else label $y' \neq y$. **[x]**
- D. It is a targeted attack.
- E. It is an untargeted attack. **[x]**

Question 13

Given a DL model $f(x;\theta)$ parameterized by θ where $f(x;\theta)$ represents the prediction probabilities of x associated with a ground-truth label $y \in \{1, \dots, M\}$, we find an adversarial example by $\boldsymbol{x_{adv} = argmin_{x' \in B_\epsilon(x)} l(f(x';\theta), y_{\neq})}$ with $y_{\neq} \neq y$.

Which statements are correct? (MC)

- A. We maximally increase the chance to predict x_{adv} with label y.
- B. We maximally increase the chance to predict x_{adv} with label $y_{\neq}$.
- C. It is a targeted attack.
- D. It is an untargeted attack.

Question 13

Given a DL model $f(x;\theta)$ parameterized by θ where $f(x;\theta)$ represents the prediction probabilities of x associated with a ground-truth label $y \in \{1, \dots, M\}$, we find an adversarial example by $\boldsymbol{x_{adv} = argmin_{x' \in B_\epsilon(x)} l(f(x';\theta), y_{\neq})}$ with $y_{\neq} \neq y$.

Which statements are correct? (MC)

- A. We maximally increase the chance to predict x_{adv} with label y.
- B. We maximally increase the chance to predict x_{adv} with label $y_{\neq}$. **[x]**
- C. It is a targeted attack. **[x]**
- D. It is an untargeted attack.

Question 14

- What are correct about adversarial training? (MC)

A. At each iteration, we use the standard data augmentation technique to augment the data.
B. At each iteration, we use an adversarial attack such as PGD to augment the data.
C. We update the model parameters to let the model predict the clean and adversarial images to their ground-truth labels.
D. The final loss consists the loss over clean examples.
E. The final loss consists the losses over clean and adversarial examples.

Question 14

- What are correct about adversarial training? (MC)

A. At each iteration, we use the standard data augmentation technique to augment the data.
B. At each iteration, we use an adversarial attack such as PGD to augment the data. **[x]**
C. We update the model parameters to let the model predict the clean and adversarial images to their ground-truth labels. **[x]**
D. The final loss consists the loss over clean examples.
E. The final loss consists the losses over clean and adversarial examples. **[x]**