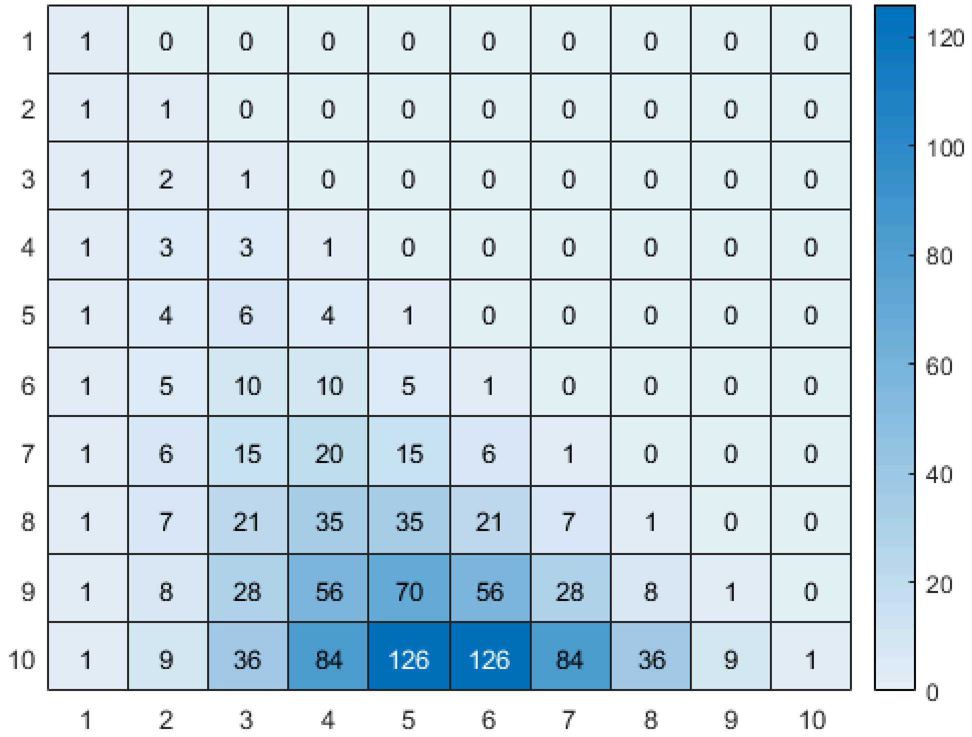
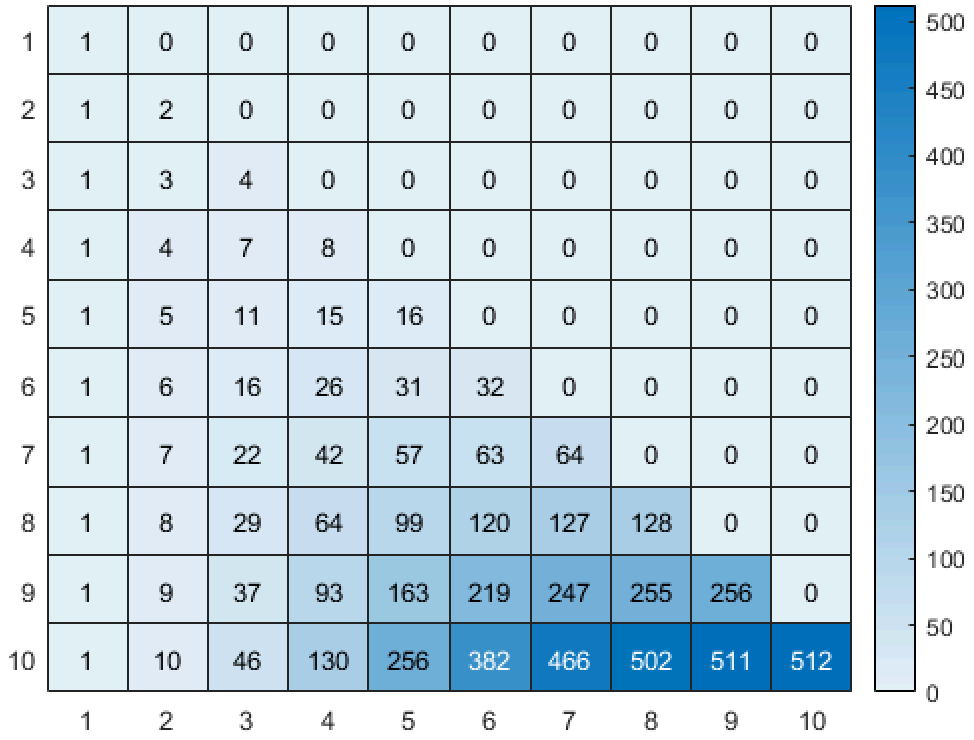
```
for i = 1:10
    cl(i) = convolution2dLayer([2,2],1,"Bias",zeros(1,1,1),"Weights",[ones(1,2,1,1);zeros(1,2,1
        "Padding",[1 0 1 0],"Name","pasc"+int2str(i));
end
layers = [imageInputLayer([10 10 1],'Normalization',"none","Name","unit_input")
cl.'
regressionLayer("Name","output")];
convPasc = assembleNetwork(layers);
inputs = zeros(10,10);
inputs(1,1)=1;
p = zeros(10,10,10);
for i = 1:10
    a = activations(convPasc,inputs,i);
    p(:,:,i) = a;
end
out = sum(p,3);
heatmap(out)
```



	1	2	3	4	5	6	7	8	9	10
1	1	0	0	0	0	0	0	0	0	0
2	1	1	0	0	0	0	0	0	0	0
3	1	2	1	0	0	0	0	0	0	0
4	1	3	3	1	0	0	0	0	0	0
5	1	4	6	4	1	0	0	0	0	0
6	1	5	10	10	5	1	0	0	0	0
7	1	6	15	20	15	6	1	0	0	0
8	1	7	21	35	35	21	7	1	0	0
9	1	8	28	56	70	56	28	8	1	0
10	1	9	36	84	126	126	84	36	9	1

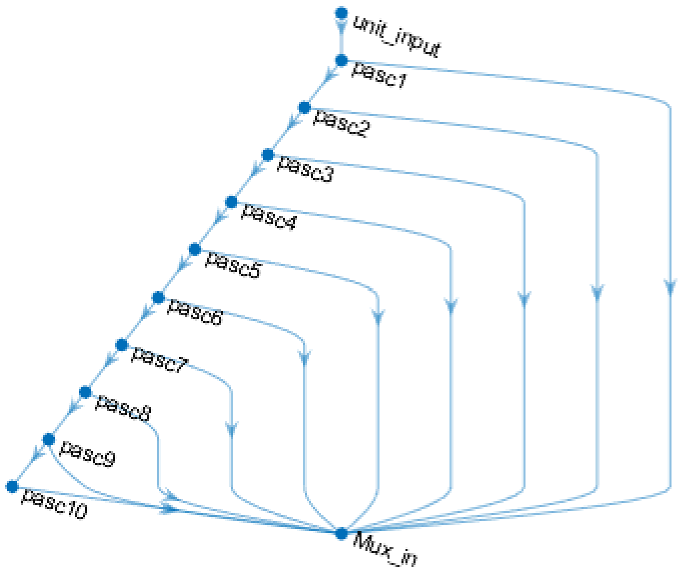
```

bernoulilayers = [imageInputLayer([10 10 1],'Normalization',"none")
convolution2dLayer([1,10],10,"Bias",zeros(1,1,10),"Weights",reshape(eye(10),1,10,1,10),...
"Padding",[0 0 9 0],"Name","Bernoulli")
regressionLayer()];
convBern = assembleNetwork(bernoulilayers);
a = activations(convBern,out,2);
% a = activations(convBern,ones(10),2);
out2 = tril(sum(a,3));
```

```
figure
heatmap(out2)
```



```
lgpasc = layerGraph(layers(1:end-1));
connector = [additionLayer(10,"Name","Mux_in")];%convolution2dLayer(1,1,"Bias",zeros(1,1,1,1),'
lgpasc = addLayers(lgpasc,connector);
for i = 1:numel(cl)
lgpasc = connectLayers(lgpasc,"pasc"+int2str(i),"Mux_in/in"+int2str(i));
end
```

```
plot(lgpasc);
axis off
```

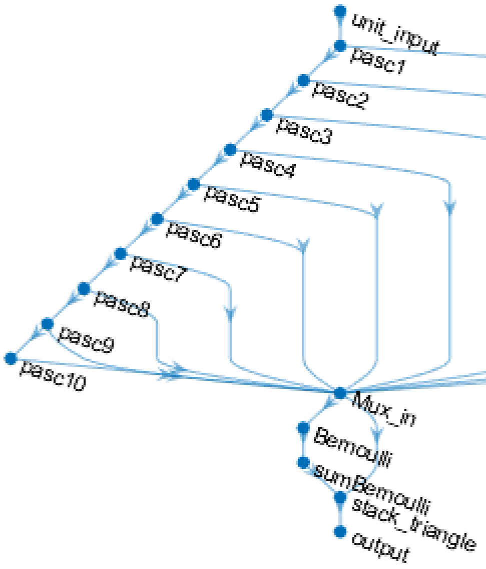


Add Bernoulli Layers.

```
lgpasc = addLayers(lgpasc,bernoulilayers(2));
lgpasc = connectLayers(lgpasc,"Mux_in","Bernoulli");
lgpasc = addLayers(lgpasc,convolution2dLayer(1,1,"Bias",zeros(1,1,1,1),"Weights",ones(1,1,10,1)
lgpasc = connectLayers(lgpasc,"Bernoulli","sumBernoulli");
```

Now add back an output layer.

```
lgpasc = addLayers(lgpasc,depthConcatenationLayer(2,"Name","stack_triangle"));
lgpasc = addLayers(lgpasc,layers(end));
lgpasc = connectLayers(lgpasc,"Mux_in","stack_triangle/in1");
lgpasc = connectLayers(lgpasc,"sumBernoulli","stack_triangle/in2");
lgpasc = connectLayers(lgpasc,"stack_triangle","output");
plot(lgpasc);
axis off
```



```
convBern = assembleNetwork(lgpasc);
p = predict(convBern,inputs);
a = activations(convBern,inputs,'Bernoulli');
```