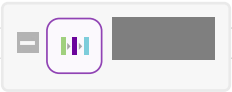
In[1]:=
```
SeedRandom[1234]; (* Ensure reproducibility *)
```

```
(* Generate training and testing data correctly *)
trainData = Table[
 Module[{A, B, C},
  A = RandomReal[{-1, 1}, {2, 2}]; (* Random 2x2 matrix *)
  B = RandomReal[{-1, 1}, {2, 2}]; (* Random 2x2 matrix *)
  C = A.B; (* Ground truth matrix multiplication *)
  <|"Input" → Flatten[Join[A, B]], "Output" → Flatten[C]|>
     (* Flattened input and output *)
 ],
 {1000}
];
```

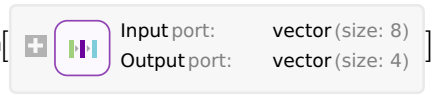
```
testData = Table[
 Module[{A, B, C},
  A = RandomReal[{-1, 1}, {2, 2}];
  B = RandomReal[{-1, 1}, {2, 2}];
  C = A.B;
  <|"Input" → Flatten[Join[A, B]], "Output" → Flatten[C]|>
     (* Flattened input and output *)
 ],
 {100}
];
```

In[4]:=
```
(* Neural network model *)
nn = NetChain[{
 LinearLayer[16], Tanh, (* Hidden layer with 16 neurons *)
 LinearLayer[4] (* Output layer for 4 elements (2x2 matrix flattened) *)
 },
 "Input" → 8, (* 8 values: 2 flattened 2×2 matrices *)
 "Output" → 4 (* 4 values: 1 flattened 2×2 matrix *)
];
```

In[5]:=
```
trainedNN = NetTrain[nn, trainData, ValidationSet → testData]
```

Out[5]=
```
NetChain[
```

```
]
```

In[6]:=
```
trainedNN = NetTrain[nn, trainData, ValidationSet → testData]
```

Out[6]=
```
NetChain[ 
  Input port:  vector (size: 8)
  Output port: vector (size: 4)
 ]
```

In[11]:=
```
testMatrices = Table[
 Flatten[Join[RandomReal[{-1, 1}, {2, 2}], RandomReal[{-1, 1}, {2, 2}]]],
 {1000}
];
```

```
nnTime = AbsoluteTiming[trainedNN /@ testMatrices][[1]]
```

Out[12]=
```
0.59884
```

In[9]:=
```
builtInTime = AbsoluteTiming[(#[[1]].#[[2]] &) /@ testMatrices][[1]]
```

Out[9]=
```
0.001133
```

In[10]:=
```
{nnTime, builtInTime}
```

Out[10]=
```
{3.4328, 0.001133}
```