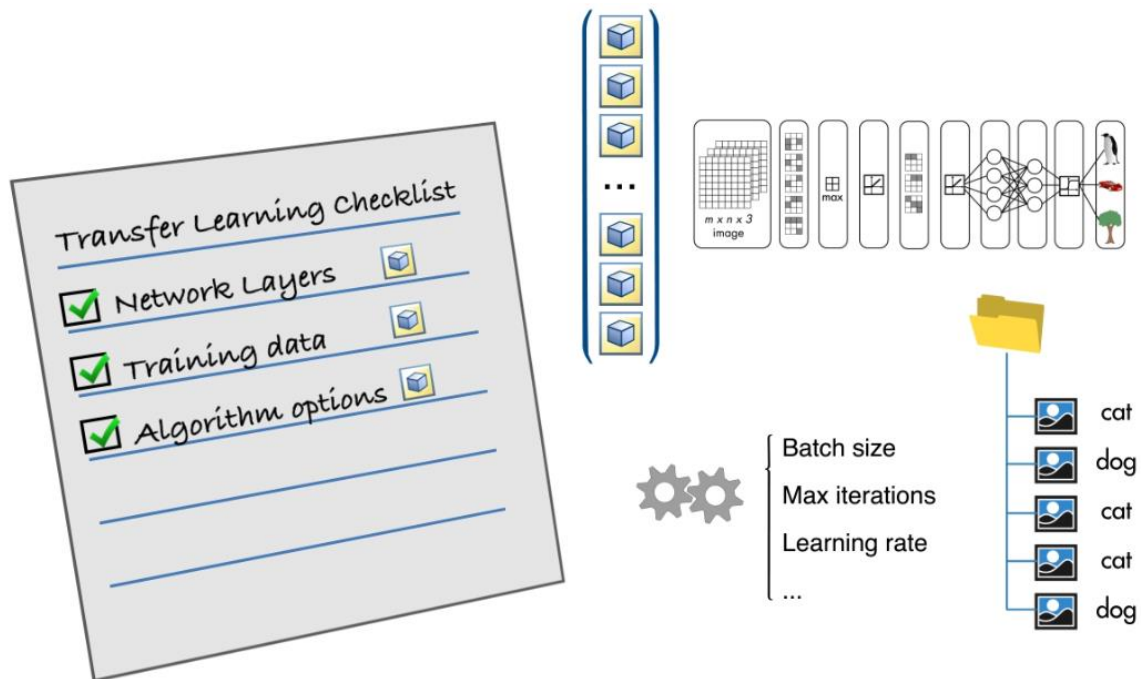


ENTRENAMIENTO DE LA RED

Realizar el Entrenamiento

Para realizar la transferencia del aprendizaje, hacen falta tres cosas. En primer lugar, necesita un arreglo de capas que represente su red. A continuación, necesita un almacén de datos para sus imágenes de entrenamiento, con la propiedad `Labels` y que guarde las etiquetas correctas para esas imágenes de entrenamiento. Y, por último, necesita una variable que contenga la configuración del algoritmo de entrenamiento.



Una vez que se tienen estos tres componentes, el entrenamiento es fácil. Solo tiene que darle estos tres elementos a la función `trainNetwork` y ponerla en marcha. Cuando haya concluido, obtendrá la red recién entrenada como salida en forma de un arreglo de capas, igual que la red de entrada, pero con pesos actualizados. De manera predeterminada, verá una pantalla de texto que muestra el progreso del entrenamiento.

La precisión es el porcentaje de imágenes de entrenamiento que la red clasifica correctamente. Lo que se busca es que se incremente durante el entrenamiento. No obstante, la precisión no mide la fiabilidad de la red en cada predicción.

```
>> newnet = trainNetwork(data, layers, options)
```

Training on single GPU.
Initializing image normalization.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.47	3.5061	7.81%	0.0010
3	10	10.31	0.7686	75.00%	0.0010
5	20	18.96	0.2371	92.19%	0.0010
8	30	27.43	0.0770	97.66%	0.0010
10	40	35.31	0.0336	99.22%	0.0010
13	50	43.17	0.0289	99.22%	0.0010
15	60	50.15	0.0104	100.00%	0.0010
18	70	56.84	0.0072	100.00%	0.0010
20	80	63.00	0.0210	99.22%	0.0010
23	90	69.37	0.0035	100.00%	0.0010
25	100	74.85	0.0027	100.00%	0.0010
28	110	81.19	0.0053	100.00%	0.0010
30	120	86.75	0.0045	100.00%	0.0010

Elapsed time is 87.899947 seconds.

La pérdida es una medida de lo lejos que se ha quedado la red de una predicción perfecta, respecto del total de imágenes del conjunto de entrenamiento. Debería tender a cero a medida que avance el entrenamiento.

```
>> newnet = trainNetwork(data, layers, options)
```

Training on single GPU.
Initializing image normalization.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.47	3.5061	7.81%	0.0010
3	10	10.31	0.7686	75.00%	0.0010
5	20	18.96	0.2371	92.19%	0.0010
8	30	27.43	0.0770	97.66%	0.0010
10	40	35.31	0.0336	99.22%	0.0010
13	50	43.17	0.0289	99.22%	0.0010
15	60	50.15	0.0104	100.00%	0.0010
18	70	56.84	0.0072	100.00%	0.0010
20	80	63.00	0.0210	99.22%	0.0010
23	90	69.37	0.0035	100.00%	0.0010
25	100	74.85	0.0027	100.00%	0.0010
28	110	81.19	0.0053	100.00%	0.0010
30	120	86.75	0.0045	100.00%	0.0010

Elapsed time is 87.899947 seconds.

Con frecuencia observará que la precisión se estanca en su valor máximo mientras que el valor de la pérdida sigue disminuyendo. Eso es bueno. Significa que la red sigue mejorando su capacidad de distinguir entre las categorías incluso si las predicciones finales no cambian.

```
>> newnet = trainNetwork(data, layers, options)
```

Training on single GPU.
Initializing image normalization.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.47	3.5061	7.81%	0.0010
3	10	10.31	0.7686	75.00%	0.0010
5	20	18.96	0.2371	92.19%	0.0010
8	30	27.43	0.0770	97.66%	0.0010
10	40	35.31	0.0336	99.22%	0.0010
13	50	43.17	0.0289	99.22%	0.0010
15	60	50.15	0.0104	100.00%	0.0010
18	70	56.84	0.0072	100.00%	0.0010
20	80	63.00	0.0210	99.22%	0.0010
23	90	69.37	0.0035	100.00%	0.0010
25	100	74.85	0.0027	100.00%	0.0010
28	110	81.19	0.0053	100.00%	0.0010
30	120	86.75	0.0045	100.00%	0.0010

Elapsed time is 87.899947 seconds.

Puede conseguir un registro de la precisión y la pérdida durante el entrenamiento solicitando una segunda salida a la función `trainNetwork`.

```
>> [newnet, info] = trainNetwork(data, layers, options)
```

Training on single GPU.
Initializing image normalization.

Epoch	Iteration	Time Elapsed (seconds)	Mini-batch Loss	Mini-batch Accuracy	Base Learning Rate
1	1	0.47	3.5061	7.81%	0.0010
3	10	10.31	0.7686	75.00%	0.0010
5	20	18.96	0.2371	92.19%	0.0010
8	30	27.43	0.0770	97.66%	0.0010
10	40	35.31	0.0336	99.22%	0.0010
13	50	43.17	0.0289	99.22%	0.0010
15	60	50.15	0.0104	100.00%	0.0010
18	70	56.84	0.0072	100.00%	0.0010
20	80	63.00	0.0210	99.22%	0.0010
23	90	69.37	0.0035	100.00%	0.0010
25	100	74.85	0.0027	100.00%	0.0010
28	110	81.19	0.0053	100.00%	0.0010
30	120	86.75	0.0045	100.00%	0.0010

Elapsed time is 87.899947 seconds.