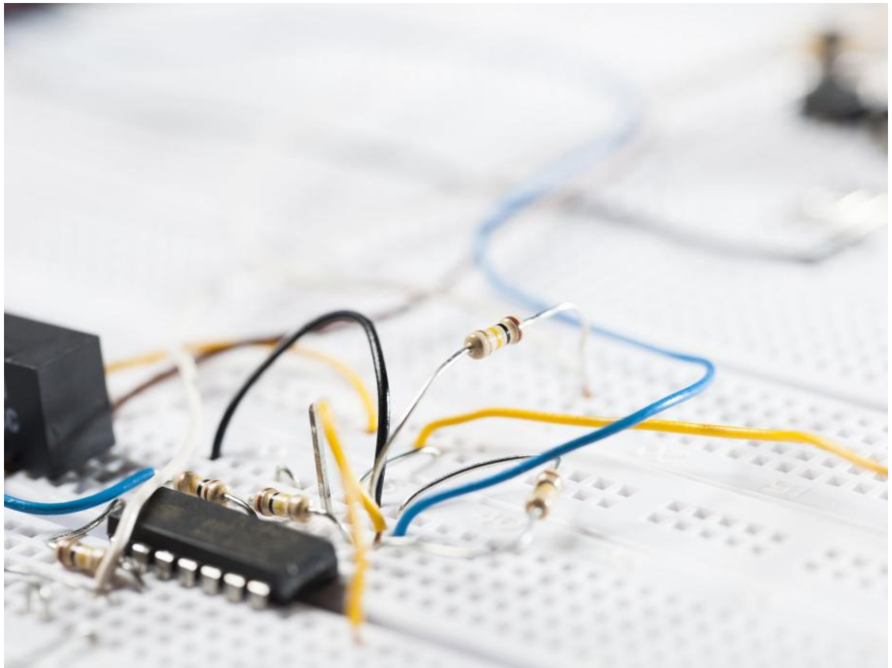
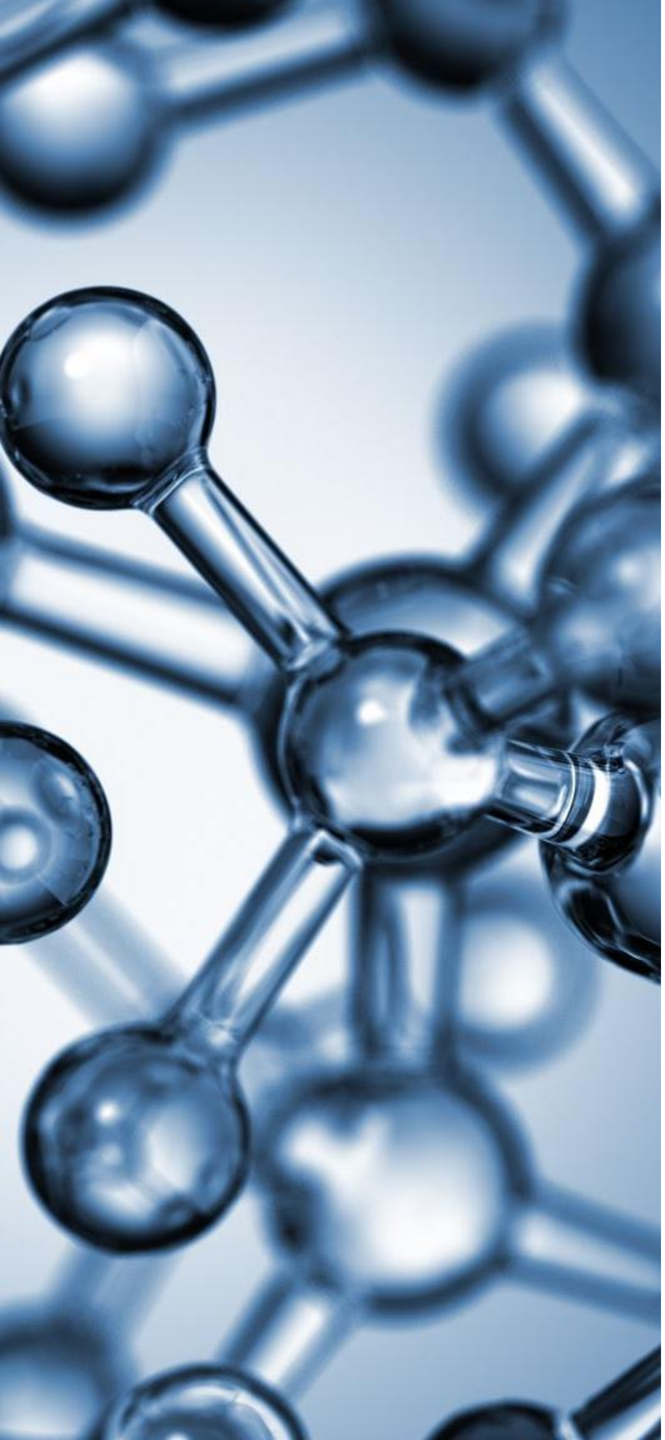
Ottimizzazione di TinyViT per Sistemi Embedded tramite Distillazione, Pruning e Quantizzazione

Autore: Davide Iannella

Progetto Deep Learning





OBIETTIVO: OTTIMIZZARE UN MODELLO TRANSFORMER (TINYVIT) PER DISPOSITIVI EMBEDDED

METODI: DISTILLAZIONE, PRUNING E QUANTIZZAZIONE.

Dataset: CIFAR-10
10 classi, immagini 32x32 RGB, ridimensionate a 224x224 per compatibilità ViT

```python
teacher = torchvision.models.resnet18(weights='IMAGENET1K_V1')
teacher.fc = nn.Linear(teacher.fc.in_features, 10)
teacher = teacher.to(device)
teacher.eval()

student = timm.create_model('vit_tiny_patch16_224', pretrained=True)
student.head = nn.Linear(student.head.in_features, 10)
student = student.to(device)

summary(student, input_size=(3, 224, 224), device=str(device))

```

```
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /root/.cache/torch/hub/checkpoints/resnet1
100%|██████████| 44.7M/44.7M [00:00<00:00, 126MB/s]
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
model.safetensors: 100% 22.9M/22.9M [00:00<00:00, 71.3MB/s]
```

Modello Teacher: ResNet-18 pretrained su ImageNet

Fine-tuned per 10 classi.

```python
teacher = torchvision.models.resnet18(weights='IMAGENET1K_V1')
teacher.fc = nn.Linear(teacher.fc.in_features, 10)
teacher = teacher.to(device)
teacher.eval()

student = timm.create_model('vit_tiny_patch16_224', pretrained=True)
student.head = nn.Linear(student.head.in_features, 10)
student = student.to(device)

summary(student, input_size=(3, 224, 224), device=str(device))

```

```
Downloading: "https://download.pytorch.org/models/resnet18-f37072fd.pth" to /root/.cache/torch/hub/checkpoints/resnet1
100%|██████████| 44.7M/44.7M [00:00<00:00, 126MB/s]
/usr/local/lib/python3.11/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
model.safetensors: 100% 22.9M/22.9M [00:00<00:00, 71.3MB/s]
```

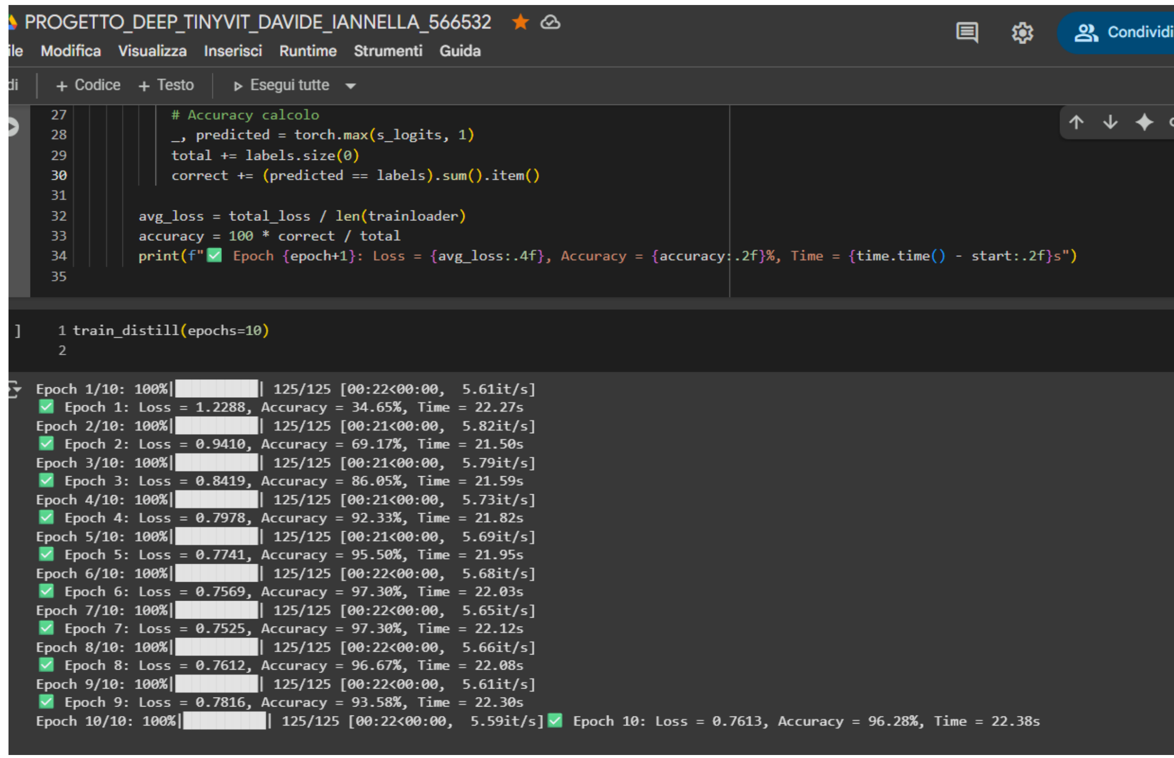
Modello Student: TinyViT (vit_tiny_patch16_224)

Piccolo ViT pre-addestrato, adattato a 10 classi.

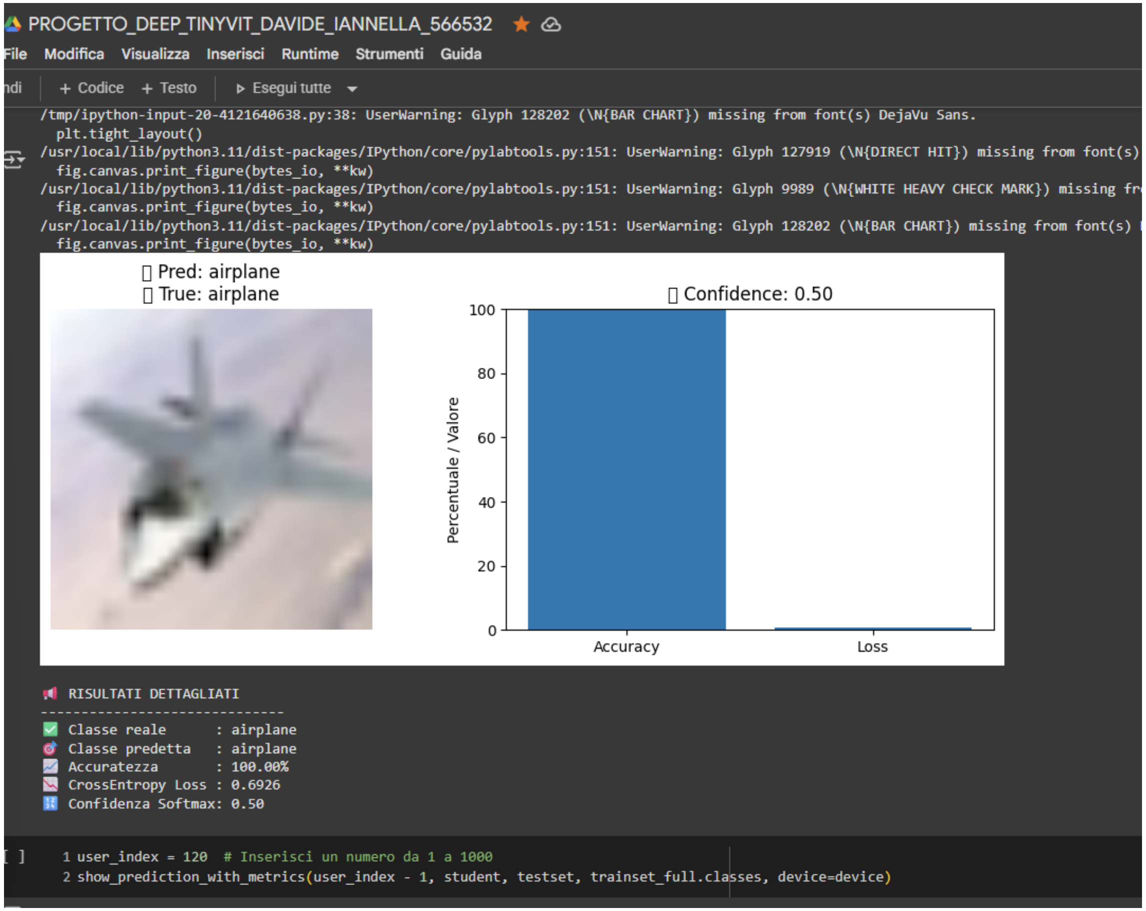
Distillazione: conoscenza trasferita dal teacher al student
Loss = KL Divergence + 0.5 * CrossEntropy.

```
loss = kd_loss(torch.log_softmax(s_logits, dim=1),
               torch.softmax(t_logits, dim=1)) + 0.5 * ce_loss(s_logits, labels)
```

Ottimizzazione: Adam, lr=3e-4, batch=32, 10 epoche su subset CIFAR10 (4000 train, 1000 test).



Risultati Distillazione: Accuratezza test > 90%, perdita ridotta, training stabile.

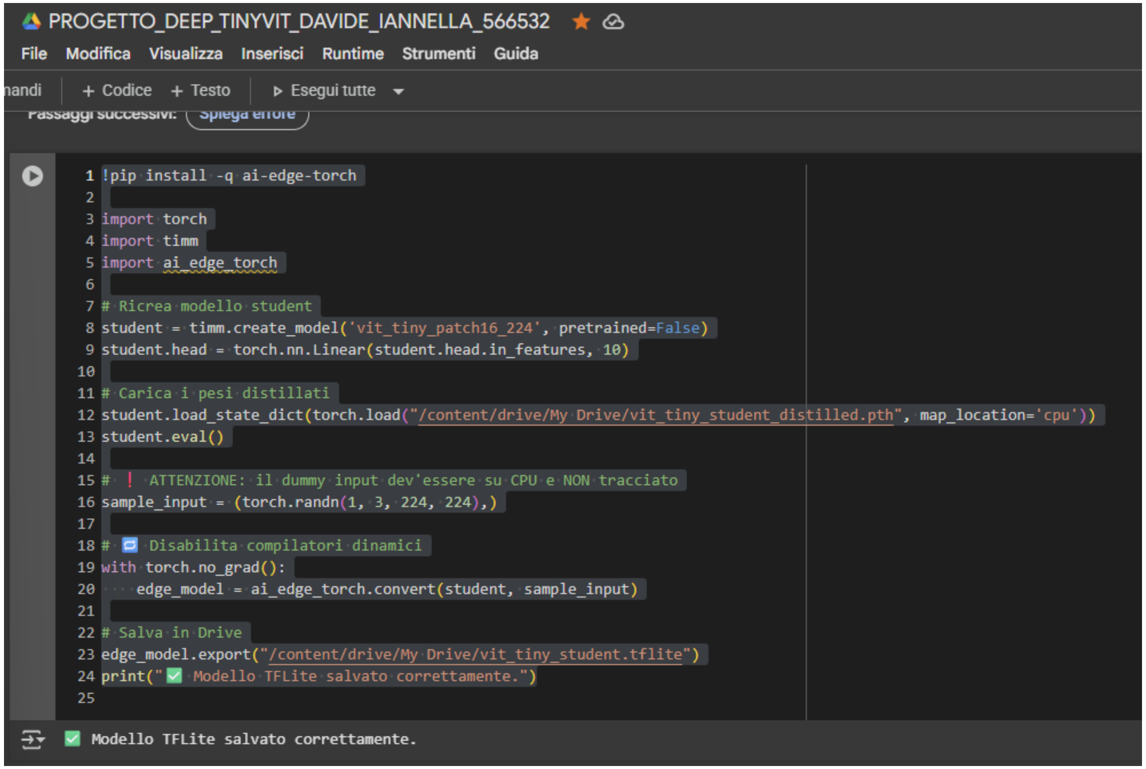


Pruning:
Disattivazione drop-out e moduli ridondanti interni a TinyViT.

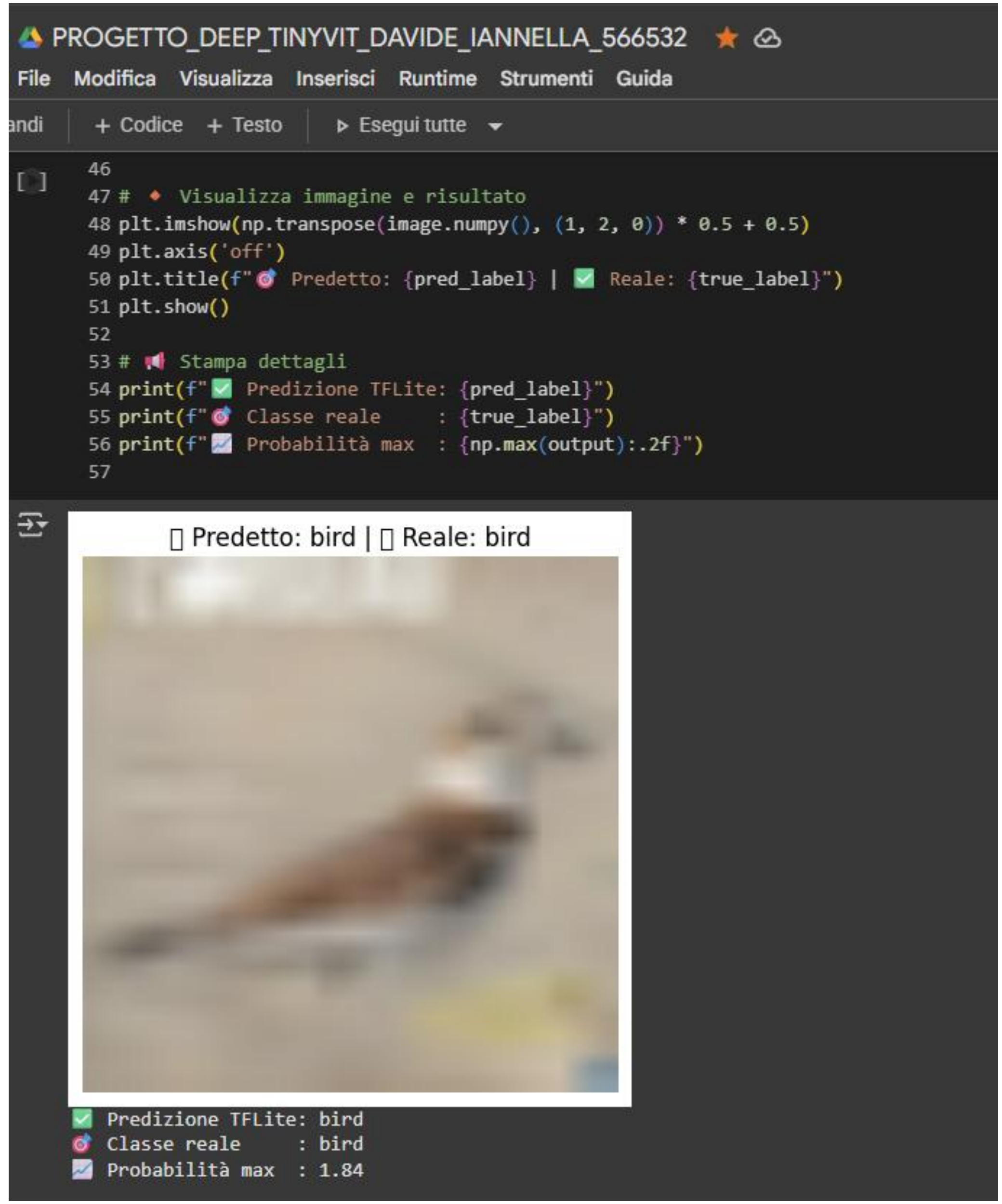
```
for name, module in student.named_modules():
    if hasattr(module, 'attn_drop'):
        module.attn_drop.p = 0.0
```

Effetto Pruning: Accuratezza conservata, modello più leggero e veloce.

Quantizzazione:
uso libreria ai-edge-torch per esportazione a TFLite
Input dummy CPU, esportazione edge_model.



Verifica Modello TFLite: accuratezza coerente, inference su CIFAR-10
Script Python + libreria TFLite + matplotlib.



Criticità Distillazione: difficile bilanciamento tra CE e KLDivLoss

Necessario tuning empirico.

Criticità Pruning: rimozione layer delicata, rischio underfitting

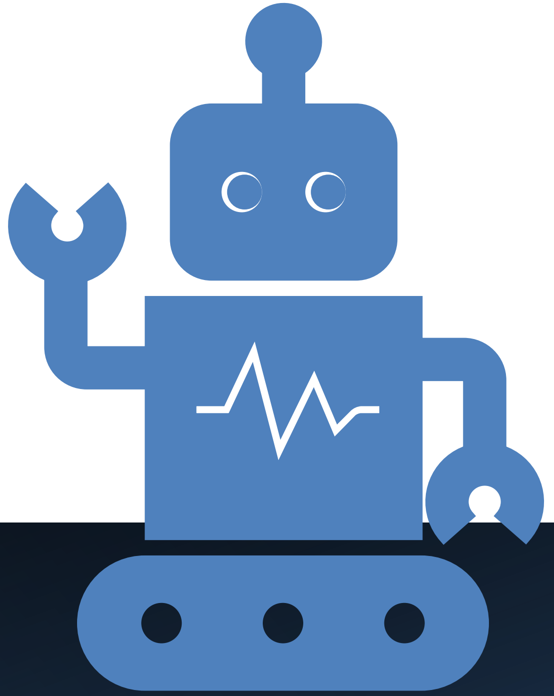
Scelta: solo dropout disattivati.

Criticità Quantizzazione: TFLite richiede CPU, incompatibilità GPU

Formati e export delicati.

Inoltre TinyViT non supporta pienamente la quantizzazione int8: il modello TFLite è float32.

Embedded Ready: modello TFLite usabile in Android, Raspberry, microcontrollori avanzati.



Struttura Codice:

1. Training distillazione
2. Pruning
3. Esportazione
4. Verifica TFLite

Futuri sviluppi:

- Addestramento full CIFAR10

- Edge deployment (Raspberry Pi)

- Quantizzazione int8.

Grazie per l'attenzione