A. Proof for Section 4.1

Proof. Consider a linear transformation \( F : \mathbb{R}^n \to \mathbb{R}^n \) converted to GNN, i.e.

\[
y_i = x_i + \sum_{j=1}^{i-1} \omega_{i,j} y_j + \sum_{j=i}^{n} \omega_{i,j} x_j, \quad \forall i \in \{1, \ldots, n\}
\]  

(1)

Suppose that there exists a pair of neurons \( y_p \) and \( y_q (p < q) \) that collapse into each other. We consider one step of gradient descent on parameter \( \omega \) with learning rate \( \epsilon \) when the input is \( x \) and the gradient on \( y \) is \( \partial L / \partial y \), i.e., \( \forall i, j, \)

\[
\Delta \omega_{i,j} = \epsilon \frac{\partial L}{\partial \omega_i} = \epsilon \frac{\partial L}{\partial y_i} \cdot \frac{\partial y_i}{\partial \omega_i}
\]

(2)

Then, we consider \( \Delta y \), which is the value difference at the same \( x \) used in gradient descent. When \( \epsilon \to 0 \), \( \forall i, \)

\[
\Delta y_i = \epsilon \frac{\partial L}{\partial y_i} \left( \sum_{j=1}^{i-1} y_j^2 + \sum_{j=i}^{n} x_j^2 + \sum_{j=1}^{i-1} \omega_{i,j} \Delta y_j \right)
\]

(3)

Since \( y_p \) and \( y_q \) collapse, \( \Delta y_p = \Delta y_q, \) i.e.,

\[
\epsilon \frac{\partial L}{\partial y_p} \left( \sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 + \sum_{j=1}^{p-1} \omega_{p,j} \Delta y_j \right) = \epsilon \frac{\partial L}{\partial y_q} \left( \sum_{j=1}^{q-1} y_j^2 + \sum_{j=q}^{n} x_j^2 \right) + \sum_{j=1}^{q-1} \omega_{q,j} \Delta y_j
\]

(4)

If \( q > p + 1 \), then we can rewrite Eq. 4 with respect to \( \Delta y_{q-1} \),

\[
\omega_{q,q-1} \Delta y_{q-1} = \epsilon \frac{\partial L}{\partial y_p} \left( \sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 \right) + \sum_{j=1}^{p-1} \omega_{p,j} \Delta y_j - \epsilon \frac{\partial L}{\partial y_q} \left( \sum_{j=1}^{q-1} y_j^2 + \sum_{j=q}^{n} x_j^2 \right) - \sum_{j=1}^{q-1} \omega_{q,j} \Delta y_j
\]

(5)

Note that the left side of Eq. 5 is a function of \( \partial L / \partial y_{q-1} \) while the right side is not. Therefore, \( \omega_{q,q-1} = 0 \). However, \( \omega_{q,q-1} = 0 \) cannot always hold after any number of gradient descent optimizations. Therefore, \( q \neq p + 1; q = p + 1 \). Thus,

\[
\epsilon \frac{\partial L}{\partial y_p} \left( \sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 \right) + \sum_{j=1}^{p-1} \omega_{p,j} \Delta y_j = \epsilon \frac{\partial L}{\partial y_{p+1}} \left( \sum_{j=1}^{p} y_j^2 + \sum_{j=p+1}^{n} x_j^2 \right) + \sum_{j=1}^{p} \omega_{p+1,j} \Delta y_j
\]

(6)

We divide the both sides of Eq. 6 with \( \epsilon \partial L / \partial y_p = \epsilon \partial L / \partial y_{p+1} \), and let \( \partial L / \partial y_p \to \infty \). Then, we have

\[
\sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 = \sum_{j=1}^{p} y_j^2 + \sum_{j=p+1}^{n} x_j^2 + \sum_{j=1}^{p} \omega_{p+1,j} \Delta y_j
\]

(7)

Eq. 7 must hold after at least one step of gradient descent on \( \omega \) with input \( x \), gradient \( \partial L / \partial y \) and learning rate \( \epsilon \), i.e.,

\[
2y_p \cdot (\epsilon \frac{\partial L}{\partial y_p} \left( \sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 \right) + \sum_{j=1}^{p-1} \omega_{p,j} \Delta y_j) + (\sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2) \cdot (\epsilon \cdot \frac{\partial L}{\partial y_p} \cdot y_p) = -\omega_{p+1,p} \Delta \left( \sum_{j=1}^{p-1} y_j^2 + \sum_{j=p}^{n} x_j^2 \right)
\]

(8)

Note that the left side of Eq. 8 is a function of \( \partial L / \partial y_p \), while the right is not. Therefore, the only solution is \( y_q = y_p = 0 \). However, these equalities will also be broken in the next step. Thus, \( y_p \) and \( y_q \) cannot collapse into each other. □

Gradually Updated Neural Networks for Large-Scale Image Recognition