Traditional MLP:

```
y = sigma(Wi*x + bi)
```

Our Lipschitz MLP:

```
import jax.numpy as jnp
def normalization(Wi, softplus_ci):
    absrowsum = jnp.sum(jnp.abs(Wi), axis=1)
    scale = jnp.minimum(1.0, softplus_ci/absrowsum)
    return Wi * scale[:,None]

y = sigma(normalization(Wi, softplus(ci))*x + bi)
```